MODELS AND METHODOLOGIES FOR GENERALIZED GROUPING PROBLEM

A Thesis submitted

in partial fulfillment of the requirements

for the degree of

MASTER OF TECHNOLOGY

by

MURARI PRASAD BARNWAL

to the

DEPARTMENT OF INDUSTRIAL & MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

JUNE, 1996

20 AUG 1986
CENTRAL LIBRARY
I. I. T. KANPUR
Acc. No. A. 122064

A122064

IME-1996-M-BAR-MOD

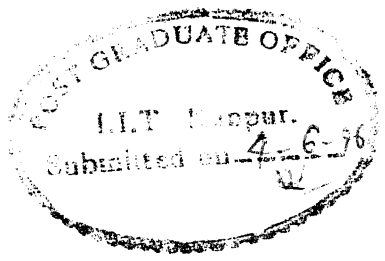
CERTIFICATE

This is to certify that the work contained in the thesis entitled **"MODELS AND METHODOLOGIES FOR GENERALIZED GROUPING PROBLEM"** by **Murari Prasad Barnwal** (Roll No. 9411413) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

June, 1996

(Kripa Shanker)

Professor
Industrial and Management Engineering
Indian Institute of Technology, Kanpur



ABSTRACT

Orientation and growth of industry towards batch production has spurred research into developing efficient means for solving its core design and operational decision problems. The result is the development of Cellular Manufacturing System (CMS), which evokes the application of Group technology. Cellular manufacturing (CM) involves processing set of similar parts on a dedicated cluster of machines. One of the problems in the design of CMS is the identification of part families and machine cells, which leads to simultaneous production of a family of related part in a flexible manufacturing system (FMS). This partitioning of the manufacturing system helps decompose the FMS planning problem into more manageable and more autonomous sub-units.

The aim of the present work is to develop solution methodology for finding part families and machine cells for some of the commonly occurring grouping scenario in CMS. The emphasis is particularly on generalized grouping problem where each part can have alternative process plans and each operation in these plans can be performed on alternative machines. Literature turns up into various techniques of solving this problem. The problem turns out to be NP-complete, hence a number of heuristic and hybrid approaches besides mathematical programming models have been developed to obtain optimal or near optimal solutions. In this work, an integer programming model named Mutually Adjacent Association Model (MAAM) has been proposed. The model is shown to be equivalent to the unit capacitated circulation network flow (UCCNF) model. The objective is to maximize (minimize) the measure of similarity (dissimilarity) between the process plans of parts and then to group them into families such that all

the parts referred by the plans selected, can be processed in their corresponding machine cells. The model does not require any knowledge about the number of part families to be formed. The solution to the problem is obtained in terms of one main loop and several sub-loops identified by using CPLEX. The concept of applying network flow in formulating the problem as a mathematical model is relatively a new approach. A simple heuristic is proposed for getting the final solution of part families with their corresponding machine cells. It appears that the proposed model (MAAM) has excellent potential to provide computationally optimal solution compared to the p-median model (PMM) by Kusiak[1987] and Agarwal's modified model.

Some measures of similarity and dissimilarity between the pairs of process plans have been proposed which are able to capture the similarity or dissimilarity between the plans with respect to processing time of operations and their sequence of occurrence.

Besides these, improvements in PMM have also been modeled which have the flexibility of either accepting the number of families in advance, or escaping from it. In the former case, termed as imposed grouping model (IMPGRP), one needs to inquire about the maximum number of parts to be kept in a family so that the computational time is the least; while in the latter case, termed as natural grouping model (NATGRP), the improved model furnishes with the number of part families by itself and one need not bother about the maximum number of parts to be kept in a family for minimum computational time to result. To obtain the final optimal solution, the heuristic proposed is applied. Computational results shows that NATGRP is much faster than p-median model (PMM) while IMPGRP is weaker in this sense.

ACKNOWLEDGMENT

I take this opportunity to convey my sincere thanks to Dr. Kripa Shanker for his guidance and help throughout the research work. The leanings from moments of formal and sometime very personal meetings with him, will channelise my thoughts. I remain indebted to him for his support and suggestions.

I express my sincere thanks to entire faculty of the IME department. Special mentions are of Prof. A. K. Mittal and Prof. T. P. Bagchi.

I express my appreciation to all the members of IME family, especially Captain, Negi and Gopal, for making my stay at IITK a memorable one.

I gratefully acknowledge my gratitude to Binod, Sanjay, Rajesh and Mohanty for their forbearance and what they are, and would be.

Murari Prasad Barnwal

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1

INTRODUCTION

Group Technology (GT) is the realization that amidst diversities many problems are similar, and that by grouping similar problems, a single solution can be found to a set of problems thus saving time and effort.

In the context of manufacturing, GT is an organizational philosophy that aims to collect manufactured parts and components into groups, based on their similarities, to facilitate their production and at the same time resulting into the effective use of manufacturing resources. This is essentially the forming of groups of parts (or part families) with similar characteristics from a design or manufacturing point of view to reduce the number of unique tasks to be dealt with. The idea behind GT is, thus, to decompose a manufacturing system into subsystems. In addition to the simplification of management control through the creation of subsystem called cell, GT leads to higher manufacturing efficiency and many other hidden benefits. A cell is essentially a group of machines dedicated to a family of related components. The cell is self contained and self managed, with performance control entrusted to the foreman who wields authority to change loading on machines, assigns operations to machines or alter the sequence of operations that he deem fit for proper load balancing within cell so long as such decisions do not contradict the overall strategies by top management.

Group Technology has probably had greatest impact on increasing manufacturing productivity than any other manufacturing concepts. It is recognized as an essential element of the foundation for the successful development and implementation of

automated manufacturing system. This is due to the contributions the cellular manufacturing concept has made to other manufacturing technologies and approaches such as Computer Aided Process Planning (CAPP), Computer Aided Manufacturing (CAM) and thereby to roboticised and flexible manufacturing systems. Recognizing the contribution that GT can make to the implementation of Computer Integrated Manufacturing System, Merchant says, "the appropriate initial step, to lay a sound foundation for the gradual evolution of a factory to full computer control, is to institute group technology cellular organization" (Choobineh, 1988). The application of GT does not depend upon the degree of automation, and may be applied in a totally automated or even in a manual production system (Choobineh, 1988).

1.1 DESIRABILITY OF GROUP TECHNOLOGY

In the competitive market, an industry involved with small lots and a large variety of parts covets to be more and more productive and at the same time more flexible to respond to the call of new and diverse products in the least time. As we know, the ideal production line for any product is mass production type which has a flow line arrangement facilitating in smooth flow production and low work-in-process (WIP) inventory. Batch and job production, however, are generally forced to adopt functional layout with a large number of general purpose machines to suit the economies of medium volume and low volume production. Consequently, such systems are afflicted by problems of long queues in front of facilities, unnecessary setup changes between parts, large amount of material handling and long manufacturing lead times.

These disadvantages and uneconomical productivity are the main concerns in the emerging business world. And the response of the newer manufacturing technologies,

both of the machines and of the tooling, to mitigate them remains incomplete without an at par improvement in organization philosophies because technological advances such as improved tools with higher cutting speed, longer tool life alone will not eliminate inefficiency since only five percent of the total time is spent in actual processing in producing a part. Moreover, technological improvements, are expensive. The organizational improvement is, however, more affordable. One such improvement in the organization of production is the application of group technology.

Group technology with the advantages (Kusiak, 1990) of reduced production lead time (20-88%), reduced work-in-process (up to 88%), reduced labour (15-25%), reduced tooling (20-30%), reduced rework and scrap materials (15-75%), reduced set-up time (20-60%), reduced order time delivery (13-136%), improved human relation and reduced paper work addresses the difficulties faced by today's manufacturing industry. Moreover the need for flexibility and just in time processing capabilities is forcing many discrete part manufacturers to consider reorganizing their facilities.

1.2 SCOPE OF PRESENT WORK

(a) Mathematical Formulation

Group technology problem has been of tremendous interest for practitioners as well as for academicians during the past three decades. Numerous analytical or heuristic methods have been developed to solve the problem of forming machines into groups and components into associated families or vice versa. But there has been little effort in developing methodologies applying network flow approach, an efficient mathematical tool. It has been pointed out by many researcher that grouping is non-polynomially (NP) hard problem. A pioneer work, perhaps the first of its kind using this approach has been

developed by Lee and Gracia-Diaz [1993] which formulates the simple grouping problem as unit capacitated circulation network with the distance measures between any two machines as the per-unit flow cost of capacitated network. Another graph theory approach based on network partitioning has been applied by Rajagopalan and Batra [1975], Vohra et. al.[1990] and Wu and Salvendy [1993]. All these studies using graph theory assume that each part has only one process plan and each operation of the process plan can be performed by only one machine. Generalized grouping concept, where a part can have more than one process plan due to the presence of alternative machine, was introduced by Kusiak [1987]. The group formation problem is to select a specific process plan for each part and group them into part families. He proposed a 0-1 linear integer programming solution known as p-median model (PMM) for this problem which assumes that number of groups is known in advance. But, this is computationally expensive even for a medium size problems (Wei and Gaither, 1990). Also, it is executed for a number of trials with different number of families to reach at the optimal solution. A more general problem environment is where each part has more than one process plan and each operation in a process plan can be performed on more than one machine. Rajmani et.al.[1990] proposed general linear and nonlinear integer programming formulations for such problem scenario which are NP hard in strong sense (Logendran et. al, 1994). Kusiak and Cho (1992) suggested an algorithm to solve part family formation in generalized grouping by solving for the cliques in a graph of nodes of process plans. This method, however, can solve the grouping problem only when the disjoint families exist which, in general, is not the case. Nagi et. al.[1990] proposed an algorithm which considers multiple process plans along with capacity of machines. The objective is to minimize the intercell movements. The algorithm solves the problems of process plan

selection and machine cell formation iteratively. Won and Kim [1994] proposed an assignment model to solve cell formation with multiple process plans. The algorithm is an iterative process using a heuristic procedure. Motivated by the work of Lee and Gracia-Diaz [1993], a new integer programming formulation named Mutually Adjacent Association Model (MAAM) using the applied concept of capacitated circulation network is being proposed to solve the generalized grouping problem. The formulation does not need the number of part families to be prespecified. However, as the formulation results into many families which are not disjoint, a heuristic procedure is proposed to merge two or more than two of them. The proposed heuristic also involves machine cell formation with the objective of minimizing the intercell movements. This is not an iterative process. Once the problem is solved it will give the optimal solution in one go. The proposed formulation along with formulation of generalized p-median method (Kusiak, 1987) and modified Kusiak's model (Agarwal, 1990) have been coded and applied to solve some examples from literature. Since the order of complexity (in terms of total number of enumerations of forming families) of the proposed formulation is very less (of quadratic order) as copperas to the p-median and Agarwal's modified model (of exponential order) it has been found to be faster than p-median formulation and gives better grouping in terms of exceptional elements. It has also been found that Agarwal's modified model is not able to solve even a middle size problem of 20-25 plans.

Improvements in the Kusiak's model (PMM) have also been proposed and tested. While in the Kusiak's model it is possible that a single part can make a family, thereby violating the GT concept itself, these improvements agree with the GT concept. These improvements are such that the solver have the flexibility in either prespecifying the number of part families or he may escape from this. They have been termed as imposed

grouping model (IMPGRP) and natural grouping model (NATGRP) respectively. NATGRP gives the number of families as its result. However, since many families resulting are not disjoint or have common machines as their requirement, the proposed heuristic is applied to reach at the optimal solution. A comparison of these improvements with the Kusiak model have also been carried out which shows that NATGRP is computationally very efficient as compared to Kusiak's model and IMPGRP.

(b) Measures of Similarity and Distance

Similarity (distance) coefficient (SC), a numerical measure of commonality (dissimilarity) between machines or parts, has been used in most of the methods known so far to solve the clustering problem. Shafer et. Al. [1990] presented a survey on this and developed a new similarity coefficient. Mosier [1989] experimented with some well known SCs designed for applying hierarchical cluster analysis to the GT machine cell formation. Very few SCs have been developed to represent the operation sequence similarity. We propose a dissimilarity coefficient, named as Operation Sequence and Machine Utilization Dissimilarity Coefficient (OSMUDC) between the process plans taking operation sequences and utilization penalty of machines into account. Some examples from the literature of grouping are solved with this dissimilarity coefficients. Results are compared with that of resulting from the application of some existing SCs on operation sequence. The coefficient simultaneously offers the most smooth machine layout in the cell, as it minimizes the material flows by considering the operation sequence. This coefficient also provides the optimum linear machine layout in a machine cell. In addition to this, two other measures related to process time have been devised. One called the Multiplicative Weighted Similarity Coefficient (MuWSC) ensures the minimization of intercell movements of parts while the other called Time Movement Distance Coefficient (TMDC)

ensures the minimization of time of the intercell movements. Another measure weighted part and time similarity coefficient (WPTSC) by making a tradeoff between the two is also proposed.

1.3 ORGANIZATION OF THE THESIS

Chapter 1 furnishes introduction of the subject of group technology. Desirability of the group technology has also been discussed.

Chapter 2 presents a brief overview of the major efforts in group technology. Moreover, discrepancies and observations are also presented.

Chapter 3 describes some selected generalized grouping techniques. Their comparison has also been performed through a table.

Chapter 4 presents the proposed mathematical model (MAAM) and the improvements in the existing Kusiak's model (PMM). Comparison of the proposed model and the improved models with Kusiak's model has been discussed on the basis of a number of criterion.

Chapter 5 devotes to the similarity and distance measures developed. Also, some existing measures parallel to them have been discussed.

Chapter 6 explains various efficiency measures to compare the results of various grouping techniques.

Chapter 7 implements and experiments with these models. It discusses and compares their results for some example problems. The graphs, charts, and tables used to analyze the solution are presented in the chapter.

Chapter 8 concludes and brings to view some further scope of research in the related field.

2

PREVIOUS WORKS : A SYNOPSIS

In this chapter a brief description of some of the major efforts in the field of group technology is presented. Since application of group technology aims at dividing the production system into subsystems, most of the research is focused on dividing the machines and components into groups. There have been numerous application of mathematical programming, heuristics, neural networks, fuzzy mathematics, expert system etc. with different kinds of inputs to suit a manufacturing environment.

A combination of inputs or a single input from the list given below is usually taken for the purpose of solving the GT problems.

- Machine-part 0-1 matrix
- Machine-part routing sequence
- Process time
- Production volume
- Alternate part routing
- Machine capacity
- Costs (Handling, operation, overtime)

The organization of description of major efforts in the field of GT is based on some classification scheme scheme as described below:

1. Rules of Thumb
2. Classification and Coding System
3. Flow Analysis
4. Approaches using Similarity/Dissimilarity Coefficients

5. Numerical Taxonomy
6. Mathematical Programming Approach
7. Heuristic Approach
8. Hybrid Approach
9. Network Approach
10. Fuzzy Mathematical Approach
11. Artificial Intelligence System

2.1 Rules of Thumb

Based on the production engineer's local knowledge of the parts and processes, some families of the components may be self evident. Families consists of components defined by name or functions e.g. pump casing, shafts and sleeves.

2.2 Classification and Coding System

In this system, the first step is to classify and code all the components using a stable system where, for every manufacturing and design attributes of the components a unique code is reserved. In the production department code wise route cards are prepared for all the components and components having similar codes are identified to form component families and thereby to form associated machine cells. In the design office the code will be useful in retrieving earlier designs of similar form. The problem of design retrieval is tackled by allocating a code number through a classification system which covers the important design features of the components.

A number of commercial (for e.g. BRISCH, CODE, MICLASS) as well as non-proprietary (for e.g. OPITZ, KC - 1) coding scheme are currently in practice.

2.3 Flow Analysis

The aim of flow analysis is that of finding the families of components and associated groups of machines for group layout, by a progressive analysis of the information contained in route cards.

(a) Production flow analysis

Burbidge [1963] deserves the credit for his pioneering work on group technology who developed production flow analysis (PFA). It involves a number of stages of analysis which may be briefly described as follows:

STAGE 1 involves classifying the machines by a number according to type, on the basis of operations that can be performed. Machines capable of performing similar operations are usually classified with the same type number and the specific needs of components for particular machines within the type are considered when allocating these machines to the groups which need them. Machines required for minor and ancillary operations are excluded from the analysis. Such ancillary machines are appropriately assigned o the required machine-component groups once these have been determined from the analysis.

STAGE 2 carries the extensive checking of the parts list and production route card information to identify and ensure correctness of the essential information for the analysis; namely, for each component, the operations to be undertaken and the machine necessary to perform each of these operations.

STAGE 3 is termed *factory flow analysis* which, as the title implies, involves a macro examination of the flow of components through the machines, which, in turn, allows the problem to be decomposed into a number of major machine-component groups.

STAGE 4 is to undertake the final *group analysis* on each of these major groups separately, rather than to attempt it on the undivided overall aggregation of machines and components. In fact, this has been the focus of attention of several researcher who take the necessary data's available as a result of the earlier stages of analysis. These data's can be of the type of input(s) listed above.

(b) Component flow analysis

Component flow analysis (CFA) was developed by El-Essawy [1971] who claimed it to be different from PFA, but the similarity of the two approaches is apparent. CFA employs three stages of analysis.

STAGE 1 objective is to consider the total component mix of the company and to identify and sort components into categories according to their manufacturing requirements. The components are sorted in order of machine requirements and sorted list is printed in two ways, firstly in the order of the number of machines required and secondly in the order of the smallest machine number involved.

STAGE 2 aims to obtain groupings of machines using the lists of sorted components and taking into account various local constraints. Rough groups are formed by using the combinations with the highest number of machines as the cores, to which other machines and components are successively added.

STAGE 3 involves a detailed analysis of the loading and flow pattern of the cells with appropriate adjustments to ensure that an acceptable design is achieved.

CFA does differ from PFA in some respects. The latter first partition the problem, whereas the former does not. The manner in which the cells are built up is also different in the two methods.

2.4 Approaches using Similarity/Dissimilarity Coefficients

The similarity coefficient approach was first suggested by McAuley[1972]. This class of methods calculate a numerical measure which describes the similarity between parts or machines. A survey of similarity and distance measures applicable to cellular manufacturing has been presented by Shafer and Rogers [1993]. These approaches are divided into two two subclasses hierarchical clustering and non hierarchical clustering.

(a) Hierarchical clustering

The basis of this method is to measure the 'similarity' between each pair of machines (parts) and then to group the machines (parts) into families based on their

similarity measurements. In most cases, the similarity measurement used is the coefficients of Jaccard which is defined for any pair of machines as 'the number of components which visit both machines, divided by the number of components which visit at least one of the machines'. The components having similar coefficients are grouped together to form a family.

McAuley [1972] used Single Linkage Cluster Analysis (SLCA) based on this similarity coefficient. This method first clusters together those machines mutually related with the highest possible similarity coefficient, then it successively lowers the level of admission by steps of predetermined equal magnitude.

Witte [1980] categorized the machines as primary, secondary and tertiary based on the interdependence of the three classes of machine types and proposed primary, secondary and tertiary similarity coefficients.

Selection of threshold value of SC in SLCA, Witte's three SCs and Rajgopalan's graph theoretic [1975] is arbitrary and it requires a certain amount of judgment. As shown by de Beer and de Witte [1978], the Jaccard coefficient fails when one of the machines processes a large number of parts which are not common to the machine pairs.

(b) Non hierarchical clustering

These methods first select seed machines and then cluster all the machines around these machine. Similarly parts are clustered around seed parts. These seed are selected from some mathematical programming approach. ZODIAC (Chandrasekharan and Rajagopalan, 1987) calculates the seed from the bipartite graph formulation, while GRAFICS (Srinivasan and Narendaran, 1992) obtains the initial seed from the assignment method.

2.5 Numerical Taxonomy

Numerical taxonomy is the widely used technique for cell formation in group technology. These methods re-arrange the rows and columns of the machine

component incidence matrix such that the 1's are brought together. Each block of 1's constitute a component family and a machine cell.

McCormick [1972] proposed Bond Energy Algorithm (BEA) which reorders the rows and columns of the matrix for the purpose of moving numerically larger matrix elements together. A measure of closeness between two component vectors or machine vectors termed bond energy is defined. Method seeks to determine permutation of rows and columns in which the sum of products of adjacent elements is maximized. This is a restricted form of quadratic assignment problem.

Graham [1976] proposed linear weighting algorithm where the weights are increased linearly. The ith row is a given a weighting of $m - i + 1$ where m is the total number of rows, and the priority ranking value is determined as the mean of the weightings of the non zero entries. Ranking values calculated this way can be found and sorted very quickly and the requirement of a very large integer representation does not arise. The major disadvantages of this linear weighting algorithm are the complicated and very confusing patterns of the intermediate results as well as the difficulty in predicting the behaviour of the procedure.

King [1980] proposed a Rank Order Clustering (ROC) technique based on numerical taxonomy. This method interprets each entry in the matrix as a binary word and computes decimal equivalent of each row and column. Rows and columns are sorted in decreasing order of the binary word values, until algorithm converges.

King and Nakornchai [1982] observed that reading entries as binary word restricted the size of the problem to less than 47 machines and 47 components because the largest integer representation in any computer was $(2^{48} - 1)$. They proposed ROC2 which is computationally efficient.

Chan and Milner [1982] proposed direct clustering which is a poor version of ROC algorithm, except that it eliminates the sensitivity of the latter to the configuration

of the initial matrix. This algorithm has the same handicaps as the ROC algorithm and uses a limited binary comparison procedure for ranking.

Kusiak [1985] discusses the computational complexities of BEA and ROCs.

Chandrshekharan and Rajagopalan [1986] observed that ROC does not provide block diagonal structure even in the case of structured matrices where such a possibility exists. They found that ROC is dependent on the initial disposition of the incidence matrix and that it does not handle exceptional elements too well. They suggested MODROC.

Khator and Irani [1987] proposed a heuristic procedure, the Occupancy Value Method (OVA), for identifying clusters in a machine component matrix created from route card data. A unique feature of this method is that it progressively develops block diagonalization starting from the northwest corner of the matrix.

Kusiak and Chow [1987] proposed cluster identification algorithm (CIA) which identifies the clusters in a binary machine-part incidence matrix provided they exist. He proposed three algorithms based on cluster identification to solve GT problem with bottleneck components and machines. The first algorithm solves an unconstrained GT problem. The second is a heuristic which considers a constraint restricting the number of machines in each cell and identifies the bottlenecks. The third named the cost analysis algorithm allows one to solve the group technology problem with known subcontracting costs. The overlapping parts are removed in such a way that the total sum of subcontracting costs is minimized.

Boe and Chang [1991] identified several deficiencies of clustering and array based sorting algorithms and suggested a clustering algorithm termed as close neighbour algorithm. In the first stage clustering of machines is performed through a 0-1 quadratic programming model maximizing the closeness of machines and in the second stage clustering of parts is done. The algorithm converges in one pass in most of the problems.

2.6 Mathematical Programming Approach

Using mathematical approaches like linear programming, 0-1 integer programming, dynamic programming, goal programming etc., the GT cell formation problem may be given concise mathematical formulation and a solution can be found for a given constraint set.

Gunasingh and Lashkari [1989] proposed two 0-1 integer programming formulations for the GT problem. The first formulation groups machines based on the compatibility of processing components, while the second formulation groups machines in order to minimize the cost of machine allocation and the cost of intercellular movement.

Gunasingh [1991] presented a non-linear 0-1 integer programming formulation to simultaneously group machines and parts in cellular manufacturing systems based on tooling requirements of the parts., tools available on the machines and the processing times. The formulation takes into account the limitations on the number of parts and machines in a group and the number of machine types available.

Boctor [1991] presented 0-1 integer programming based formulation for cell formation with the objective of minimizing the exceptional elements. This model allows the designer to control the cell sizes.

Kusiak [1987] developed a integer programming problem. He considered presence of alternative process plans and suggested a 0-1 integer program based p-median model (PMM) to determine the optimal process plan and assignment of parts to machine cells. Incorporating process plans will result in an improved quality of part families and machine cells.

Song and Hitomi [1992] considered the part routing, part volume and cell size. They formulated the problem in two steps : first, a QAP formulation to minimize the number of parts within a cell and the second, a graph formulation for minimization of intercell part flows. Global optimum is obtained using branch and bound method.

Rajmani [1992] developed an mixed integer programming model to minimize the sum of total discounted cost and set up cost. He considered dependent set up time, cost for switching the product, cost and capacity of machine.

Rajmani [1994] developed three integer models for the same problem. First model assigns machines to parts using any clustering algorithm. Second model assigns machines to known part families. Part families are obtained by part attributes. Third model obtains part families and machine groups simultaneously considering demand time and resource constraint. He considered alternate machines and alternate routings for part, and cost and capacity of machine.

Wei and Gaither [1990] developed a 0-1 integer programming model to assign machines and parts to cells simultaneously. This model can solve larger cell formation problems than earlier approaches because it uses far fewer decision variable than earlier p-median formulation. The objective of this model is to minimize the cost of manufacturing exceptional part types outside the cell subject to machine capacity constraints.

Kumar [1986] formulated a joint grouping and loading problem as a multistage multi-objective model. The min-max approach to multiobjective optimization is used to obtain a compromise solution.

Shtub [1989] showed that the generalized cell formation problem due to Kusiak [1987a] is equivalent to the generalized assignment problem (GAP).

Agarwal [1990] investigated p-median formulation of Kusiak [1987] and GAP of Shtub [1989] and improved formulations requiring lesser number of variables and constraints.

Choobineh [1988] proposed a two-stage procedure to design a cellular manufacturing system. In the first stage, part family is determined by similarity coefficient based algorithm. In the second stage an integer programming formulation is used to form machine cells with an objective of minimizing production cost and the cost of acquiring and maintaining machine tools.

Srinivasan [1990] proposed an assignment model for GT. The similarity coefficient matrix for machines is solved as an assignment problem from which initial cells are identified. Component families are similarly identified and they are merged using a set of rules.

Dahel and Smith [1993] proposed a multiojective model for grouping parts and machines simultaneously into prespecified number of cells. The objective of the model is to minimize inercell moves as well as to maximize the processing flexibility by maximizing the number of machines in each cell subject to machine capacity and cell size constraints.

2.7 Heuristic Approach

Since even the relaxed version of the grouping problem is NP-complete, it is unlikely to ever get the optimal solution to the original problem efficiently. Therefore, the cell formation procedures found in the literature are predominantly heuristic which can find good solutions in a reasonable amount of computational time. The heuristic can be problem specific or general purpose.

Lenstra [1972] pointed out that the clustering problem is equivalent to two travelling salesman problems. The computational complexity of the two problems is non-polynomially-complete (NP) for large problem.

Shtub [1989] showed that the simple cell formation problem is equivalent to the generalized assignment problem (GAP) which is NP-hard as the partitioning problem (NP- complete) is reducible to GAP (Fisher et.al., 1986).

Ballakur et. al. [1987] have proposed a within-cell utilization heuristic for designing cellular manufacturing systems. The distinguishing feature of this heuristic is its consideration of several practical criteria such as within-cell utilization, work load fractions, maximum number of machines that are assigned to a cell, and the percentage of operations of a parts completed within a cell.

Vannelli et. al. [1986] discussed a method of identifying the minimal number of bottleneck cells which, when dealt with through either duplication of machines or

subcontracting of parts, will result in perfect part machine grouping with no overlap. The polynomially bounded algorithm used in the analysis are oriented towards finding minimal cut-nodes in either partition of the bipartite part-machine graph.

Askin et. el. [1991] proposed a hamiltonian path approach to reordering the part machine incidence matrix for cell formation. Similarity coefficients have been used to form a distance measure for machines. Using the distance matrix, a heuristic solution to the associated travelling salesman problem (TSP) was obtained. Finally using the heuristic tour determined for TSP a heuristic solution to the associated hamiltonian path problem was obtained.

Harahalakis et. al. [1990] proposed a simple twofold heuristic algorithm capable of minimizing intercell material movement by considering the sequence of operation. The first stage is a bottom-up aggregation procedure to minimize the 'normalized inter cell traffic'defined as the ratio of the inter cell movement between any two cells under consideration and the number of machines included in these cells. The second stage is aimed at improving the solution obtained from the first stage, in which the significance of a machine to a cell is validated.

Logendran [1990] proposed a model which incorporates both intracell and intercell moves. Logendran [1991] considered sequence of operations into account in the design of cellular manufacturing system. This model includes the impact of cells in evaluating intercell moves. The model also aims at maintaining a targeted utilization of work station.

Balasubramanium [1993] described an algorithm which helps the formation of machine groups. The algorithm first considers several possible cells. It then develops a similarity matrix to reflect the component's routeing similarity with that of the arrangement of the facilities in the cell. He developed similarity coefficient considering production volume and operation similarity. The machine grouping is done heuristically by considering the cost of handling (using SC) and the cost of

idle time of machine (using machine loads) and overtime simultaneously. The same machine type is allowed in more than one cell if necessary.

Venugopal and Narendran [1992] presented an algorithm based on simulated annealing to solve the machine-part grouping problem. The algorithm partitions the machine-part incidence matrix in such a way that the total intracell load variation is minimal.

Genetic algorithms (GA) a general adaptive search method which mimics the process of natural evolution were developed by Holland [1975]. Since then GAs have been applied to diverse areas of optimization. Its success in combinatorial optimization attracted the researchers of GT to apply GA for solving grouping problems. Venugopal and Narendran [1992] seem to be the first who applied GA for solving grouping problems. They formulated the grouping problem as a bi-criterion mathematical program and solved it using GA. Computational results show that the proposed algorithm can be successfully applied to large scale grouping problems.

2.8 Hybrid Approach

The survey till now has shown that certain cell formation methods employs both heuristic and mathematical procedures in different stages. Such methods are called hybrid approaches.

Steudel and Ballakur [1987] proposed a two-stage heuristic in which the first stage employs a dynamic programming procedure to generate an optimum chain of machines in which the sum of bonds between machines in the chain is maximized. The second stage is a heuristic, that partitions the chain to form machine cells subject to cell size restrictions.

Nagi et. al. [1990] addresses two problems in the presence of multiple routeing process plans for a part : (a) process plan selection which has been solved by formulating it as a LP problem, (b) cell formation, is solved through a heuristic.

The method also considers the sequence of operations and production data such as production volume, processing time and machine capacity.

Choobineh [1988] employed a two stage procedure. First employs a similarity measure based on part operations and operation sequences and attempts to identify part families using a heuristic procedure. The second stage formulates as an integer programming which specifies the type and number of machines in each cell and the assignment of part families to the cells.

Askin and Chiu [1990] incorporated costs of inventory, machine depreciation, machine setups and material handling costs into a mathematical programming formulation. A heuristic graph partitioning procedure is then applied to solve it.

2.9 Network Approach

The network approach to GT problem can be divided into two categories.

2.9.1 Network partitioning approach

Machines and parts are treated as nodes and machining time or SC is put on the arcs connecting them. These models aim at obtaining disconnected subgraphs for identifying cells and part family.

Rajagopalan and Batra [1975] suggested a graph partitioning approach which used cliques of machine-graph as a means of grouping machines. He used the Jaccard SCs on the arc.

Vanneli and Kumar [1986, 1987] have developed algorithm to identify minimal bottleneck machines to be duplicated and components to be subcontracted to obtain non-intersecting part families and cells for a given matrix.

Vohra et. al. [1990] preesented a non-heuristic network approach to form machine cells with minimum intercellular interactions. The machine part matrix containing machine times is represented as a network which is subsequently partitioned by using a modified Gomory-Hu algorithm to find a minimum cut between nodes of machines.

Wu and Salvendy [1993] proposed a modified version of Vohra's work and for larger problems it gives a good solution but not the optimal solution.

Graph-decomposition is a difficult combinatorial task and the above mentioned methods do not guarantee optimality.

2.9.2 Network flow approach

Lee and Diaz [1993] have described a simple grouping problem as a maximum flow network problem. Each machine is represented by two nodes and each arc has three values (U,δij,L) assigned where δij is the SC between machine i and machine j and U and L are the upper and lower limit on the flow respectively.

There has been no work found to solve generalized grouping problem using this approach.

2.10 Fuzzy Mathematical Approach

Most of the methodologies for cell formation discussed earlier assume that the information about processing costs, processing times, part demand, etc. is precise (deterministic) but in reality it is not. Fuzzy clustering provides a solution to such problems. Relatively few papers such as Xu and Wang [1989] and Chu and Hayya [1991] have addressed the issues of vagueness in cell formation problems.

2.11 Artificial Intelligence (AI) Based Techniques

Recent development in AI techniques for solving decision making problem in various fields have influenced manufacturing and group technology, too. A brief review of the AI based approaches for simple grouping is given below.

2.11.1 Expert system and knowledge based system

Kusiak [1988] first proposed EXGT-S, the tandem architecture of knowledge based system for simple grouping problem. It has two parts, namely expert system and heuristic clustering algorithms. Each partial solution generated by heuristic algorithm is evaluated by the expert system which modifies search direction of algorithm. The approach presented allows one to take advantage of the user's

production expertise, and can be used to determine machine cells and part families. However, the EXGT-S is unable to detect the machine chaining problem. Chow and Hawaleshka [1993] presented another knowledge based system which can be used for identifying machine cells and part families and helps in rectifying some of the problems commonly encountered in a EXGT-S system.

2.11.2 Neural network based methodologies

The neural network is a new algorithmic approach which has been the subject of intensive study of mathematician, statisticians, physicists, engineers and computer scientists. Its parallel processing capability, learning ability, and less storage space requirements have made it popular for solving large scale real life problems. Kusiak [1991] proposed adaptive resonance theory based neural network developed by Carpenter and Grosberg to solve the GT problems.

Karpathi and Suresh [1992] used Carpenter and Grosberg neural network. Since this clustering method utilizes binary-valued inputs and it can be trained without supervision. Experiment on larger data sets (10000 x 100) revealed that the method results in the identification of clusters with fast execution times.

Chu [1993] applied a neural network approach based upon a competitive learning paradigm for designing part families and machine cells. The procedure can effectively obtain optimal cluster result.

2.12 Observations

The most important point to be noted is that optimum result of GT problem may or may not be possible to get because of its NP-complete nature. Basically all the research papers have moulded the original GT problem according to some constrained environment. Original GT problem is not just the grouping on the basis of machine part matrix, rather it should get integrated with material requirement planning, just-in-time, market demand structure, assembly operations and many more which influences the operation of the production shop.

It is because of its NP-complete nature that it is not even possible to predict whether an optimal solution to a problem exists

Based on literature survey presented following observations can be made :

- Less efforts have been made to incorporate flexibility and its measure in the design of group technology.
- Network flow approach, an efficient mathematical tool, has not been applied to the generalized grouping problems.
- Less work on facilities design (layout) along with cell formation with required sequence to facilitate smooth flow production.
- In most of the approaches production volume and the machine loads were not considered. In some case it was considered only after the cell formation was over.
- In all the SC methods involving production volume and workloads, SC is independent of production volume.

3

SOME TECHNIQUES FOR GENERALIZED GROUPING

This chapter presents the details of some of the techniques which are used to solve the GT problem where every part has a set of process plans. However, all of them can not be equally optimal in respect of various manufacturing yardsticks like total time taken by the plan, number of types of machine required, skill requirement, and of course the most important being quality control. But with reference to the concept of GT, one might even settle to choose a sub-optimal plan in order to confine a part specific cell. This may be preferred to reduce queuing times, transportation times, costs, and WIP, and to increase productivity. Another reason for alternate process plans is the existence of functionally similar workcentres i.e. alternative machines. In a functional layout, this is not a consideration, because the part can be routed to any such available workcentre. For example, a part requires a turning operation. Our turning facility consists of two identical lathes. The part could use either lathe when it is routed to the turning facility. On the other hand, in a GT environment, manufacturing cells usually consists of functionally dissimilar workcentres; the two lathes are likely to be placed in different cells. In this case, we would prefer to route the part to the lathe in its corresponding cell. Thus, there is a need to identify each workcentre as a specific one. This unique identification of functionally similar workcentres lead to the existence of multiple routeing.

Thus, the cell formation problem in the presence of alternate routeing is to choose one of them to determine the part families with most similar route of other parts. Then subsequent machine allocation to these families produces the machine cells. We have selected some of the representative mathematical approaches which are described in the following sections.

3.1 Generalized p-median method (PMM) [Kusiak, 1987]

3.2 Modified p-median model [Agarwal, 1990]

3.3 Shtub's Assignment Model [Shtub, 1987]

3.4 Maximum Clique Algorithm (MCA) [Kusiak and Cho, 1992]

3.1 Generalized p-median model (PMM)

Kusiak [1987] introduced the concept of generalized grouping. Based on this concept the part-machine matrix is replaced by the machine process plan binary matrix, with the basic and alternate process plans. The objective is to form groups of the process plans such that the total similarity (distance) defined in some suitable manner is maximized (minimized), and only one process plan from the several alternatives of a part is selected in the group formation. This model requires the desired number of part families in advance.

Notation and definitions

Indices

i, j : process plans

k : part

m : machine

Parameters

M : the total number of machines

K : the total number of parts

p : the maximum number of process families required

q : the total number of process plans over all parts

F_k : set of process plans for part n

$a_{i,m}$ = 1 if process plan i requires machine machine m

= 0 otherwise.

s_{ij} = similarity between process plan i and j

The elements of the similarity coefficient matrix [s_{ij}] are computed as follows.

(a) Similarity between two process plans of the same part

$$s_{ij} = -\infty \tag{3.1}$$

(b) Similarity of a process plan with itself

$$s_{ij} = 0 \tag{3.2}$$

(c) Similarity between two process plans of different parts

$$s_{ij} = \sum_{m=1}^{M} d(a_{im}, a_{jm}), \quad \forall i \in F_k ; \forall j \notin F_k \text{ and } k = 1, ..., K \tag{3.3}$$

Decision Variables

A process family is identified by the index of one of its member plans. The p required families, therefore, will assume their indices from the range 1 to q.

For i, j = 1, ..., q, the following decision variable is defined:

x_{ij} = 1 if process plan i belongs to process family j

= 0 otherwise.

Formulation

The objective of the generalized p-median model is to form the process plan families by maximizing the value of similarity measure among them.

$$\text{Maximize} \quad \sum_{i=1}^{q} \sum_{j=1}^{q} s_{ij} x_{ij} \tag{3.4}$$

subject to

(I) Selection of only one process plan for a part

$$\sum_{i \in F_k} \sum_{j=1}^{q} x_{ij} = 1 \qquad k = 1, ..., K \tag{3.5}$$

(II) Formation of maximum required number of groups

$$\sum_{j=1}^{q} x_{jj} \leq p \tag{3.6}$$

(III) Assignment of processs plans to groups

$$x_{ij} \leq x_{jj} \qquad i, j = 1, ..., q \tag{3.7}$$

(IV) Binary decision variables

$$x_{ij} = 0 \text{ or } 1 \tag{3.8}$$

3.2 Modified p-median model [Agarwal, 1990]

Agarwal [1990] modified the constraint of assigning the process plans to group in generalized PMM. He observed that the number of assignments along the diagonal of the similarity coefficient matrix $[s_{ij}]$ is equal to p (see eqn. 3.6). Thus, the remaining (K-p) assignments have to be made corresponding to some non-diagonal elements of the matrix in a manner such that these (K-p) assignments appear in only those p columns or p rows that represent the process families and have assignments at their cells corresponding to the diagonal of the matrix. These observations have been used to simplify the process plan assignment constraint (i.e. eqn. 3.7) of the PMM. The simplified constraint in place of equation (3.7) is in the form of following inequality

$$\sum_{i=1}^{q} x_{ij} \leq (k - p + 1) x_{jj} \qquad j = 1,.....,q \qquad (3.9)$$

The idea behind the constraint is that out of assignments of K plans (K-p) assignments have to be made from the non-diagonal side and since to any process family of p process families to be made, at the most (K-p+1) plans can be assigned when only one plan (i.e. the index plan) is assigned to other (p-1) families.

3.3 Shtub's Assignment Model

The grouping problem has been formulated as a generalized assignment problem (GAP). The structure of GAP as described by Shtub is as follows.

Notation and definition

I	set of tasks
J	set of agents
$i \in I$	index set of tasks
$j \in J$	index set of agents
c_{ij}	the cost incurred if task i is assigned to agent j
$l_{ij} \geq 0$	the amount of a resource required by agent j to perform task i

$a_j \geq 0$ the minimum amount of the resource that may be expended by agent j

$b_j \geq 0$ the maximum amount of the resource that may be expended by agent j

$x_{ij} = 1$ if task i is assigned to agent j

$= 0$ otherwise.

Formulation

Minimize

$$\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \tag{3.10}$$

subject to

$$a_j \leq \sum_{i \in I} l_{ij} x_{ij} \leq b_j \quad \text{for all } j \in J \tag{3.11}$$

$$\sum_{j \in J} x_{ij} = 1 \quad \text{for all } i \in I \tag{3.12}$$

The objective function minimizes the cost of assigning tasks to agents. Constraint set (3.11) ensures minimum and maximum usage of the resource by each agent, and set (3.12) ensures that each task is assigned to exactly one agent. He also showed that the GAP formulation is equivalent to generalized PMM by taking $c_{ij} = -s_{ij}$, and $c_{jj} = -M$, a negative number where $M > \max\{s_{ij}\}$.

3.4 Maximum Clique Algorithm (MCA)

Kusiak and Cho [1992] proposed this algorithm to solve the part family formation problem in presence of basic and alternative process plans of parts. This procedure uses the similarity coefficient approach. A similarity coefficient between process plans is defined and a branch and bound algorithm is proposed to cluster the process plans according to their similarity value.

The algorithm represents the process plan similarity matrix by a transition graph and finds the maximum clique of the graph which is considered as a part family. Similarly all the part families are generated by branch and bound procedure.

Definition

s_{ij} = similarity coefficient between two process plans i and j of the process plan -machine matrix $[a_{im}]$ is given by

$$s_{ij} = 1 \text{ if } a_{im} \geq a_{jm} \text{ or } a_{im} \leq a_{jm} \quad \text{for all machine m} \qquad (3.13)$$

$$= 0 \quad \text{otherwise.}$$

The value of similarity coefficient $s_{ij} = 1$ indicates that the one process plan is a subset of another process plan. So the parts represented by the process plans do not require different type of machines and can be put in a single family.

Algorithm

The algorithm works on the branch and bound procedure. Similarity coefficient between process is calculated and stored in a similarity matrix which is used to construct a transition graph. Every node of a graph represents a process plan and an arc between two nodes suggests a similarity coefficient of 1 between process plans. The disconnected nodes have the similarity of 0 value. All the closed tour with the maximum number of arcs between nodes known as maximum clique is found out and treated as a process plan families. After deleting the parts corresponding to these process plans the similarity matrix is revised and the procedure is repeated by the branch and bound procedure resulting into disjoint part families.

Steps

STEP 1: Begin with the generalized matrix $[a_{im}]$ between process plans and machines and set the level L = 0.

STEP 2: Initialization, Transform matrix $[a_{im}]$ to similarity matrix [sij].

STEP 3: Represent the current similarity matrix with transition graph.

STEP 4: Determine maximum cliques in the transition graph by using any available heuristic. Assign level number L= L+1 to each clique and branch on a clique with the maximum total sum of node degree.

STEP 5: Delete from [sij] the rows and columns corresponding to parts that have process plans included in clique selected.

STEP 6: Exclude from further consideration a clique which produces larger number of process families than the current solution. [Fathoming]

STEP 7: Stop when the current solution is acceptable, otherwise backtrack to nearest unfathomed node at level L and repeat from Step 3.

3.5 Discussions

In some practical cases maximum number of families (p) in PMM might be the result of experience or planning requirement. If the number of process families p is not known one can start with a small number of process families $p = p_o$ and solve the model. If the quality of process families is not satisfactory p is increased by 1 and the model is solved again till a satisfactory result is obtained.

Table 3.1 presents the details related to the number of variables and constraints used by the above models. From the table, it can be seen that for all practical grouping problems, the number of constraints in the Shtub GAP model will be less as compared to that of PMM. However, the number of the decision variables will always be more for GAP. The Agarwal's modified method requires the least number of constraints compared to both the models PMM and GAP and involves the same number of variables as for PMM. Thus, the Agarwal's modified method would contain for sure, the least number of constraints and decision variables.

Table 3.1 Number of constraints and variables for different models

	Kusiak's PMM	Agarwal's modified PMM	Shtub's GAP
Number of constraints	$q^2 + K + 1$	$q + K + 1$	$4q + K + 1$
Number of variables	q^2	q^2	$q^2 + 3q + 2K$

The most computationally complex steps in MCA are step1 and step3. The computational time complexity of step 1 is $O(M^2q)$ where q is the total number of rows in the incidence matrix i.e. total number of process plans. The complexity of step 3 is dominated by clique problem which is NP complete. It's complexity depends upon the heuristic used. So this algorithm is the most simple algorithm available for the part family formation in presence of alternative process plans.

4

SOME PROPOSED MATHEMATICAL MODELS

Bearing the problem environment discussed in the previous chapter, we propose a new integer programming model for the generalized grouping problem. This problem has been formulated as an integer programming model (PMM) by Kusiak [1987] and as an assignment model (GAP) by Shtub [1987]. Shtub [1987] also showed that his assignment model is equivalent to the Kusiak's integer programming model, however there are lesser number of constraints but larger number of variables (Table 3.1). In the literature of generalized grouping Kusiak's model is supposed to be the most representative mathematical model. It has been found during the survey that most of the techniques developed till now have been compared with PMM for their performance evaluation. It is so because it is the first technique in this field and also quite efficient. But for larger problem the performance worsens in respect of time spent in getting the solution. It is because of the complexity of the model being of exponential order. It also requires the number of process families to be specified in advance. This is what we call imposed grouping. This will surely affect the process of natural grouping. The satisfactory result is achieved by playing with the number of process families. His assumption that for the process family formation at least one process plan is necessary is also not a practical assumption from the point of view of GT concept.

(1) The model proposed is inspired from the Lee and Diaz [1993] who applied network flow approach to solve simple grouping problem where each part has a single plan. He performed the grouping of machines by minimizing the dissimilarity between machines. The proposed model forms the part families in the presence of alternate process plans for a part. Of the several process plans of each part only

one process plan is selected while forming the part families with the objective of maximizing (minimizing) the similarity (distance) among them. It does not require the number of part families to be specified in advance, rather it results in natural grouping i.e it gives the number of part families and the part families by itself. The solution methodology used for the proposed model has been discussed with an example problem. Interpretation of the resulting solution which forms process families is made through the loop formation The model is also described to be equivalent to the Unit Capacitated Circulation Network Flow (UCCNF) problem.

(2) An improvement in the Kusiak's model has also been proposed. The modified formulations have the flexibility of either not specifying the number of process families in advance or can be specified if so desired by adding the corresponding constraint in the formulation Besides it has certain other advantages over both the Kusiak's model and its modified version by Agarwal. A comparison of Kusiak's model, Agarwal's modified model, the proposed model and the improved Kusiak's models has been made on the basis of the number of constraints, number of variables, order of the complexity, the way of grouping and the agreement with GT concepts.

(3) A simple heuristic is proposed for the merging of two or more process families and the assignment of machines to the process families resulting into machine cell formation

In the following sections we describe the proposed model and the improvement in the Kusiak's model

4.1 PROPOSED MATHEMATICAL MODEL : MUTUALLY ADJACENT ASSOCIATION MODEL (MAAM)

Definitions and Notation :

The following definitions and notation are used for the model development:

M : the total number of machine types

N : the total number of parts

q : the total number of process plans over all parts

L : the total number of process families

Indices:

i, j : process plans

k : part

l : process family

m, n : machines

c : machine cell

F_k : set of process plans for part k

$|F_k|$ = the total number of process plans for part k

Parameters:

a_{im} = 1 if process plan i requires machine m

= 0 otherwise

x_{ij} = 1 if plan i associates with plan j

= 0 otherwise

$u_{ml} = \sum_{i \in l} a_{im}$, the usage factor for machine m in the process family l

d_{ij} = distance measure between process plan i and j

s_{ij} = similarity measure between process plan i and j

It may be noted that process families ultimately corresponds to the part families.

4.1.1 Distance (Similarity) Parameter Computation

A distance coefficient between process plan i and j is defined as (Lee, 1981)

d_{ij} = distance measure between process plan i and j

$= \sum \delta (a_{im}, a_{jm})$; $i \in F_k, j \notin F_k$ $m = 1$ to M

$= 0$; if $i = j$

$= +\infty$; when $i, j \in F_k$. (4.1)

where

$\delta (a_{im}, a_{jm}) = 1$; if $a_{im} \neq a_{jm}$

$= 0$; otherwise.

Similarly,

s_{ij} = similarity measure between process plans i and j

$= \Sigma \delta (a_{im}, a_{jm});$ $\quad i \in F_k, j \notin F_k, \quad m = 1 \text{ to } M$

$= 0;$ $\quad$ when $i = j$

$= -\infty;$ $\quad$ when $i, j \in F_k.$ (4.2)

where

$\delta (a_{im}, a_{jm}) = 1 ;$ if $a_{im} = a_{jm}$

$= 0 ;$ otherwise.

Decision variables:

x_{ij} = 1; if plan i associates with plan j

= 0; otherwise.

The square matrix of distance $D = [d_{ij}]$ or similarity coefficient $S = [s_{ij}]$ between the process plans thus created following the above definitions is used in the proposed model. A distance coefficient is an indicator showing the degree of dissimilarity between a pair of process plans. The process plan-machine incidence matrix is used for this purpose. In this matrix each machine is identified uniquely by a column number and each process plan by a row number indexed consecutively over all parts. The matrix A of size q xM is represented by $A = [a_{im}]$, i = 1,....., q, m = 1,....., M.

4.1.2 Integer Programming Formulations

An integer programming formulation for the process family formation in the presence of alternate process process plans is presented below. This formulation assumes that the number of process families to be formed is not prespecified and that each of the process families to be formed has at least two process plans of two different parts which, in fact, is in agreement with the group technology concept.

Objective function

The objective function is to minimize (maximize) the total dissimilarity (similarity) between the process plans and can be written as:

$$\text{minimize} \quad \sum_{i=1}^{q} \sum_{j=1}^{q} d_{ij}\, x_{ij}$$

$$\text{or} \quad \text{maximize} \quad \sum_{i=1}^{q} \sum_{j=1}^{q} s_{ij}\, x_{ij} \tag{4.3}$$

Constraints

1. Selection of one and only one process plan of a part for association with only one of the plans of some other parts, i.e.

$$\sum_{i \in F_k} \sum_{j \notin F_k} x_{ij} = 1; \quad \text{for all parts } k = 1, \ldots\ldots, K \tag{4.4}$$

2. Selection of a plan not belonging to the part considered (i.e. part k) which associates with a plan of the part considered. This plan and the plan selected by the constraint 1 (eq. 4.4) belong to different parts other than the part considered.

$$\sum_{j \notin F_k} \sum_{i \in F_k} x_{ji} = 1; \quad \text{for all parts } k = 1, \ldots\ldots, K \tag{4.5}$$

3. part considered the process plans selected by the above two constraints are the same so that a Ensuring that for the group formation results.

$$\sum_{i \in F_k} \sum_{j \notin F_k} x_{ij} + \sum_{j \notin F_k} \sum_{i \in F_k} x_{ji} = 0 \text{ or } 2; \quad \text{for plan } i = 1, \ldots\ldots, q \tag{4.6}$$

It ensures that for a part either a plan will get selected (right hand side = 2), or will not get selected (right hand side = 0).

4. Integrality of the decision variables

$$x_{ij} = 0 \text{ or } 1; \quad \text{for plans } i,j = 1, \ldots\ldots, q \tag{4.7}$$

The constraints are explained for example problem given in Table 4.1 and are shown in Appendix A.

Further, the total number of constraints = K of (4.4) + K of (4.5) + q of (4.6)

= 2K + q.

4.1.2.1 Equivalence of MAAM to the unit capacitated circulation network flow (UCCNF) model

We discuss first (section 4.1.2.1a) the UCCNF model for the machine cell formation proposed by Gracia and Diaz [1993] and then we propose the procedure for constructing the circulation network for the case of generalized grouping problem (section 4.1.2.1b). Gracia and Diaz [1993] solved the problem for the case of simple grouping problem, each part having its basic process plan only. Later we will show how the proposed integer programming model for generalized grouping can be viewed as an UCCNF model for minimum cost flow problem (MCFP) by analyzing its variables, objective function and the constraints (section 4.1.2.1c).

4.1.2.1a UCCNF model for machine cell formation (Lee and Diaz , 1993)

Lee and Diaz [1993] proposed the unit capacitated circulation network flow (UCCNF) approach to solve the machine cell formation problem for the case of simple grouping, each part having their basic plan only. He solved the machine cell formation problem in place of part family formation. The argument is that *generally* the number of machines is less than the number of parts. The problem is formulated as the minimum cost flow problem (MCFP). The distance parameter between the machines is used as the per unit flow cost.

A capacitated circulation network G = (V, E), where V is the set of nodes and E the set of arcs, can be formulated on the basis of the distance parameters between the machines. In the network, each machine m is represented by two nodes m_a and m_b connected by a directed arc (m_a, m_b). It is assumed that the number of cluster is unknown and that each cluster has at least two machines. The procedure followed by Lee and Diaz [1993] to construct the network is described below:

1. Create 2M nodes represented by m_a, m_b, m = 1, ..., M. Arbitrarily, let node 1_b be the source node and 1_a the sink node of G = (V, E).

2. Structure the rest of the network according to the following rules. Each arc is assigned three values U, L and C to indicate an upper bound on its flow, a lower bound on its flow and the per unit flow cost, respectively.

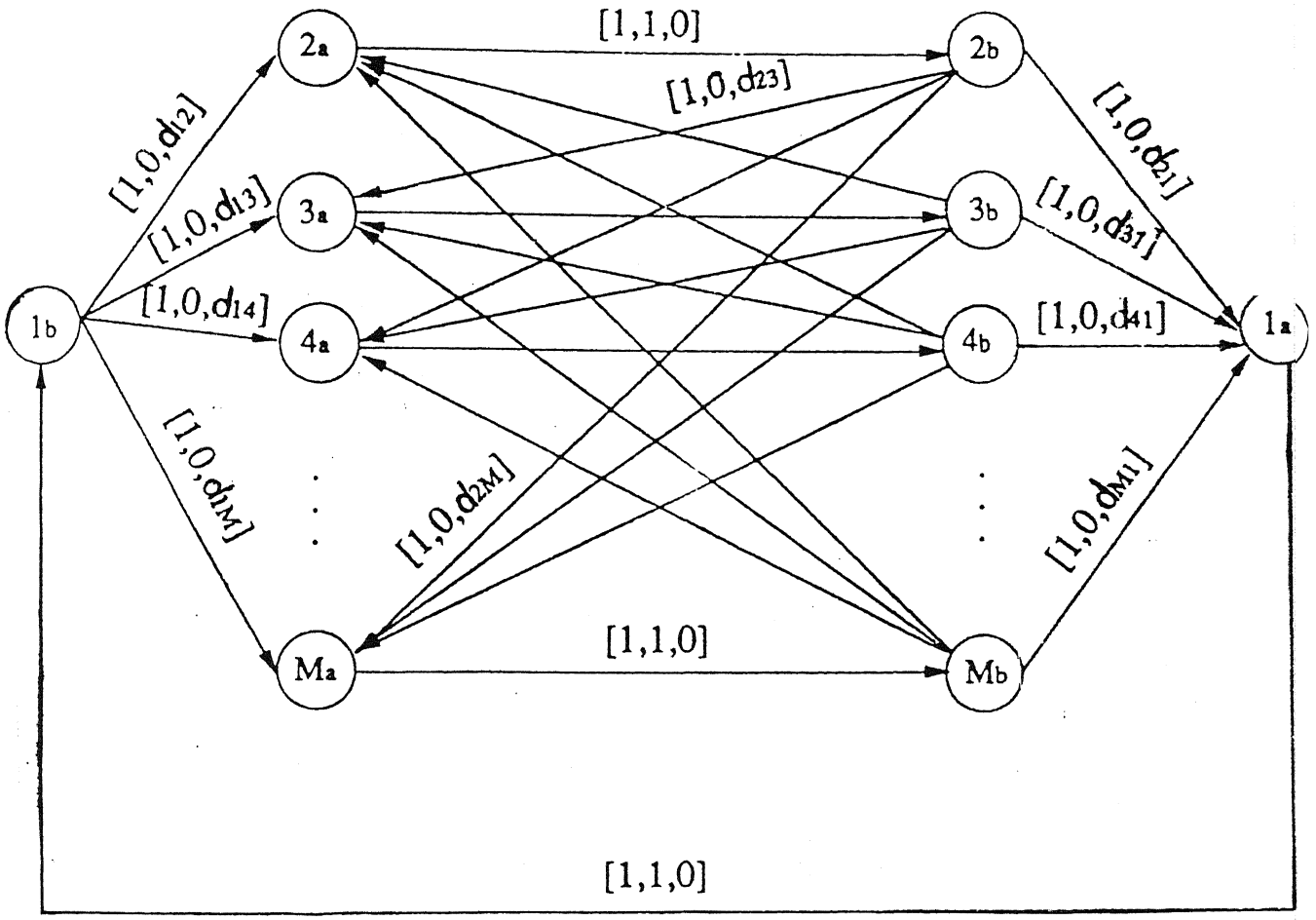


FIGURE 4.1: A Network Configuration of an M-Machine Clustering Problem

2.1 For each node m_a, $m \neq 1$, create an arc directed from m_a to m_b with capacity-cost triplet defined by [U, L, C] = [1, 1, 0].

2.2 For each node m_b, m =2, ..., M, create (M-2) arcs directed from m_b to m_a, m = 2, ..., M , with a capacity-cost triplet [U, L, C] = [1, 0, d_{mn}] , $m \neq n$.

3. For each of the nodes 1_a and 1_b, create arcs directed from 1_b to 1_a with capacity -cost triplet [U, L, C] = [1, 0, d_{1n}], $n \neq 1$, n =2, ..., M ; and arcs directed from m_b ($m \neq 1$) to 1_a with capacity-cost triplet [U, L, C] = [1, 0, d_{mn}], m =2, ..., M.

4. Create a return arc (1_a , 1_b) with capacity cost triplet [U, L, C] = [1, 1, 0] to transform the network G = (V, E).

A circulation network representation for an M-machine clustering problem is shown in Fig. 4.1. The flow on arc (m_a, m_b) is always 1 so that each node m_b can supply 1 unit of flow to one destination node n_a ($n \neq m$), by the flow conservation condition. The network solution is equivalent to minimizing the total sum of dissimilarity values between the machines used in the clustering problem. There is a formation of one main loop which is a closed loop from the source node (1_b) to the sink node (1_a). Additionally, sub-loop(s) is (are) also formed which corresponds to a closed chain not containing both the source node and the sink node. These closed loops correspond to the machine cells.

4.1.2.1b Circulation network construction for generalized grouping

The distance parameter or the negative similarity parameter defined in section 4.1.1 is used as the per-unit cost of flow and the corresponding network flow problem then becomes the minimum cost flow problem (MCFP).

In order to construct the circulation network for process family formation in generalized grouping problem following procedure is proposed.

1. Create two nodes represented by Pk_a and Pk_b for each part k , k = 1,....., K. Also, arbitrarily let node $P1_b$ be the source node and $P1_a$ the sink node of the resulting network G = (V, E).

2. Create two nodes represented by i_a and i_b for each process plan i = 1, ..., q.

Rest of the network is structured according to the following rules. Each arc is assigned with capacity-cost triplet values [U, L, C] to indicate an upper bound on its flow, a lower bound on its flow and the per unit flow cost, respectively.

3. For each node Pk_a, $k \neq 1$, create an arc directed from Pk_a to Pk_b with capacity-cost triplet defined by [U, L, C] = [1, 1, 0].

4. From each node Pk_b, create arcs directed to each i_b, $i\hat{I}F_k$ with a capacity-cost triplet [U, L, C] = [1, 0, 0] .

5. For each node i_a , $i\hat{I}F_k$, $k \neq 1$, create an arc directed from i_a to i_b with capacity cost triplet [U, L, C] = [1, 0, 0] .

6. For each of the nodes i_b, $i\hat{I}F_k$, k = 1,.....,K create (q - $|F_k|$) arcs directed from i_b to each node j_a, $j\ddot{I}F_k$, with a capacity-cost triplet [U, L, C] = [1, 0, d_{ij}].

7. For each of the nodes i_b, create an arc directed from i_b to j_a with capacity-cost triplet [U, L, C] = [0, 0, B], $i,j\hat{I}F_k$, k = 1,....., K where 'B' is a very large positive number; or, equivalently, *do not* create any arc directed from i_b to j_a, $i,j \in F_k$.

8. For each node i_a, $i \in F_1$, create an arc directed to the sink node Pk_a with capacity-cost triplet [U, L, C] = [1, 0, 0].

9. Create a return arc (Pk_a, Pk_b) with capacity-cost triplet [U, L, C] = [1, 1, 0] to transform the network G = (V, E) into a circulation network.

A circulation network representation for a K-part and q-process plans clustering problem is shown in Fig. 4.2. The arcs (Pk_b, Pk_a) has the capacity-cost triplet [U, L, C] = [1, 1, 0] which indicate that the flow through these arcs are always 1 so that each node Pk_b can supply 1 unit of flow on one of the arcs (Pk_b, i_a), $i \in F_k$, therby the node i_b connected to i_a by arc (i_a, i_b) can supply 1 unit of flow to the destination node j_a, $j \notin F_k$, by the flow conservation condition. The cost of shipping 1 unit of flow from each node i_b, $i \in F_k$ to each destination node j_a, $j \notin F_k$ is the dissimilarity coefficient d_{ij}. Thus the network solution is equivalent to minimizing the total sum of distances between the plans involved in clustering along with selecting only one plan of each part.

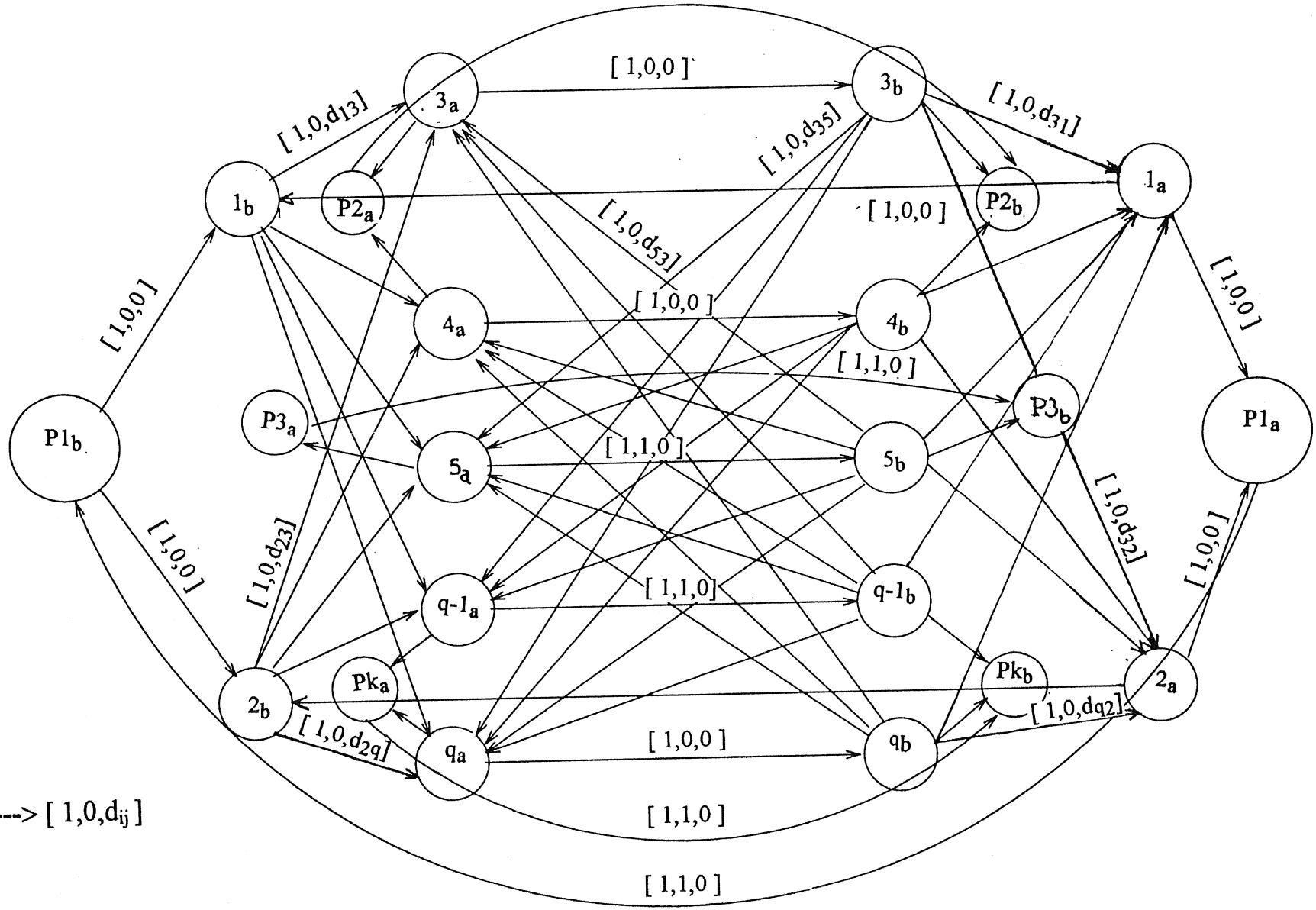


FIGURE 4.2 A Network Configuration of K part-q plan Generalised Clustering

4.1.2.1c Equivalence of the proposed model (MAAM) to the UCCNF model

Let the flow $f(i_b, j_a)$ on arc (i_b, j_a), $i \in F_k$, $j \notin F_k$ be represented by x_{ij} and the flow $f(i_a, i_b)$ on arc (i_a, i_b) by x_{ii}. Then the objective function of the proposed model which minimizes the total dissimilarity between the plans transforms into minimum cost flow problem for the circulation network constructed in the previous section. The transformed objective function is

minimize

$$\sum_{i \in F_k} \sum_{j \notin F_k} dij\, f(i_b, j_a) + \sum_{i,j \in F_k, i \neq j} d_{ij} f(i_b, j_a) + \sum_{i \in F_k} d_{ii} f(i_a, i_b)$$

Since $f(i_b, j_a) = 0$ and $d_{ii} = 0$, finally the objective function is

minimize

$$\sum_{i \in F_k} \sum_{j \notin F_k} d_{ij} f(i_b, j_a) \tag{4.8}$$

The constraints are as follows:

(I) constraint (4.4) becomes

$$\sum_{i \in F_k} \sum_{j \notin F_k} f(i_b, j_a) = 1; \text{ for all part } k = 1, \ldots\ldots, K \tag{4.9}$$

(II) constraint (4.5) becomes

$$\sum_{i \in F_k} \sum_{j \notin F_k} f(j_b, i_a) = 1 \tag{4.10}$$

(III) constraint (4.6) becomes

$$\sum \sum f(i_b, j_a) + \sum \sum f(j_b, i_a = 0 \text{ or } 2$$

$i \in F_k, j \notin F_k$ for all nodes representing process plan. (4.11)

Studying the network reveals that each node i_a, $i \in F_k$ can either supply 2 units of flow or 0 unit of flow depending upon whether its corresponding i_b is supplied 1 unit of flow or not from the node Pk_b. If it gets the supply, then corresponding i_a will supply 1 unit of flow on arc (i_a, i_b) so that i_b in turn can supply 1 unit of flow (constraint 4.9) to its destination node j_a, $j \notin F_k$ and the other unit of flow to node Pk_a for which it receives unit supply from node j_b (constraint 4.10) and becomes j_b's destination node. Thus the constraints (4.4) , (4.5) and (4.6) together indicate the flow conservation conditions for the nodes of the network. Constraint (4.7) is for lower and upper bound on the flow.

Solution of the network results into formation of closed loops. Among these one will be the main loop containing both the nodes i_b and i_a of plan $i \in F_1$ which are connected with source node $P1_b$ and the sink node $P1_a$ respectively as $(P1_b, i_b)$ and $(i_a, P1_a)$. Rest of the loops termed as sub-loops do not contain both the source and the sink node. These loops corresponds to the process families.

4.1.2.2 Solution methodology

The model formulated through equation (4.1) to (4.5) is a 0-1 integer LP problem which can be solved by any procedure solving 0-1 integer LP model e.g. LINDO or CPLEX. Solution of the model results into some non-zero variables which is then interpreted as forming some loops. The number of loops gives the number of process family formed.

We give here an illustration. The table below shows the parts and their process plans. Each part may have more than one process plan. The resulting solution is in terms of the reference number of the plan.

Example problem

The problem with 5 parts (K = 5), a total of 11 process plans (q = 11) and 4 machines (M = 4) is shown in the Table 4.1.

Table 4.1 : Input Data for the Example Problem

part k	plan number of part k	plan reference number i	machines			
			1	2	3	4
1	1	1	0	0	1	1
1	2	2	0	1	0	1
1	3	3	1	1	0	0
2	1	4	0	1	1	0
2	2	5	1	0	1	0
3	1	6	1	0	0	1
3	2	7	0	1	0	1
4	1	8	1	0	0	1
4	2	9	1	0	1	0
5	1	10	0	0	1	1
5	2	11	1	0	0	0

The distance coefficient matrix $D = [d_{ij}]$ calculated for the above problem and used in the objective function is shown in Table 4.2.

Table 4.2 : Distance matrix for the example problem

plans	1	2	3	4	5	6	7	8	9	10	11
1	0	∞	∞	2	2	2	2	2	2	0	3
2	∞	0	∞	2	4	2	0	2	4	2	3
3	∞	∞	0	2	2	2	2	2	2	4	1
4	1	2	2	0	∞	4	2	4	2	2	3
5	2	4	2	∞	0	2	4	2	0	2	1
6	2	2	2	4	2	0	∞	0	2	2	1
7	2	0	2	2	4	∞	0	2	4	2	3
8	2	2	2	4	2	2	0	0	∞	2	1
9	2	2	4	2	0	2	4	∞	0	2	1
10	0	2	4	2	2	2	2	2	2	0	∞
11	3	3	1	3	1	1	3	1	1	∞	0

After solving the problem using the proposed formulations, MAAM (eqs. 4.3 - 4.7) by CPLEX we get the following results.

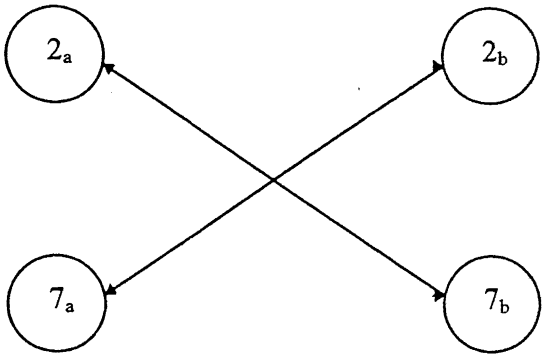
$x_{27} = 1$, $x_{72} = 1$, $x_{59} = 1$, $x_{911} = 1$, $x_{1105} = 1$; and all other x_{ij}'s are zero.

From this we can infer that plan reference number 2 and 7 form a loop and plan number 5, 9, and 11 form another loop (see Fig 4.3). Thus two process families {2,7} and {5,9,11} are formed. The result exactly matches with that by Kusiak's model. Correspondingly part families are: {1,3} and {2,4,5}.

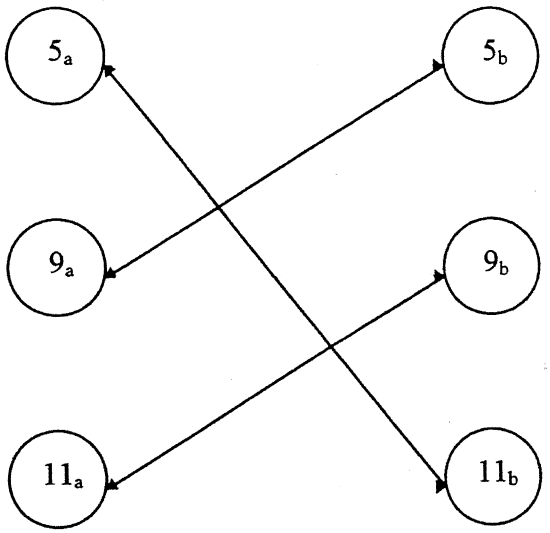
4.2 IMPROVEMENTS IN THE KUSIAK'S FORMULATIONS

Referring to the Kusiak's p-median model [1987] as discussed in section 3.1 the constraints of assigning process plans to a process family (eqn 3.7) is

$$x_{ij} \leq x_{jj} \qquad \text{for all } i, j \qquad (4.12)$$



Main loop



Sub loop

FIGURE 4.3 : Loops formation for the example problem

Through this constraint Kusiak postulated that *at least one* plan should be selected for the process family formation. This implies that even one process plan can make a process family. But it is not a very practical assumption from the point of view of GT. concept. In fact, in GT problem we do not usually consider and include that process plan which is disjoint of all other plans since there is no similarity of the manufacturing attributes of this plan with any other plans. Hence, it is argued in agreement with the GT concept that *at least two* process plans of two different parts should be selected out of the several plans available for a process family formation. Accordingly, the above constraint is modified as:

$$\sum_{i=1}^{q} x_{ij} = 0; \quad \text{if } x_{jj} = 0.$$

$$> 1; \quad \text{if } x_{jj} = 1. \tag{4.13}$$

The integer programming formulation for the above constraint in terms of inequalities is given as,

$$\sum_{i=1}^{q} x_{ij} \leq B x_{jj}. \qquad \text{for all } i, j \tag{4.14}$$

and

$$\sum_{i=1}^{q} x_{ij} \geq 2 x_{jj}. \qquad \text{for all } i, j \tag{4.15}$$

where B is the maximum number of parts desired to be kept in a family.

The constraints (3.6) of the Kusiak's model for the formation of 'p' required number of process families is

$$\sum_{j=1}^{q} x_{jj} \leq p. \tag{4.16}$$

The advantages of the constraints (4.14) and (4.15) in place of constraints (3.8) are of three fold.

1. This modification will not require the constraints (4.16) for the formation of a prespecified required number of process families. Thus it results into natural grouping.

2. If one wishes to use the constraint (4.16) when the number of process families is given in advance it can be included as well. Thus it provides the flexibility between imposed grouping and natural grouping.
3. Besides these the number of the constraints is also decreased in comparison to that of Kusiak's model.

4.2.1 Improved Models

Based on the above improvements and observations, the following two models are proposed.

(1) Imposed Grouping model (IMPGRP)

This results by replacing the constraints (3.7) by the constraints (4.14) and (4.15) in the Kusiak's model (section 3.1). The complete formulation is presented as

The objective of the model is to form the process plan families by maximizing the value of similarity measure among them.

$$\text{Maximize} \quad \sum_{i=1}^{q} \sum_{j=1}^{q} s_{ij} x_{ij} \tag{4.17}$$

subject to

(I) Selection of only one process plan for a part

$$\sum_{i \in F_k} \sum_{j=1}^{q} x_{ij} = 1 \qquad k = 1, \ldots\ldots, K \tag{4.18}$$

(II) Formation of maximum required number of groups

$$\sum_{j=1}^{q} x_{jj} \leq p \tag{4.19}$$

(III) Assignment of process plans to groups

$$\sum_{i=1}^{q} x_{ij}^{..} \leq B x_{jj}^{..} \qquad \text{for all } i, j \tag{4.20}$$

and

$$\sum_{i=1}^{q} x_{ij}^{..} \geq 2 x_{jj}^{..} \qquad \text{for all } i, j \tag{4.21}$$

(IV) Binary decision variables

$$x_{ij} = 0 \text{ or } 1 \tag{4.22}$$

(2) Natural Grouping model (NATGRP)

This results by eliminating the constraints (3.6) and replacing the constraints (3.8) by the constraints (4.14) and (4.15) in the Kusiak's model (section 3.1). The complete formulation is presented as :

The objective of the model is to form the process plan families by maximizing the value of similarity measure among them.

$$\text{Maximize} \quad \sum_{i=1}^{q} \sum_{j=1}^{q} s_{ij} x_{ij} \tag{4.23}$$

subject to

(I) Selection of only one process plan for a part

$$\sum_{i \in F_k} \sum_{j=1}^{q} x_{ij} = 1 \qquad k = 1, \ldots\ldots, K \tag{4.24}$$

(II) Assignment of process plans to groups

$$\sum_{i=1}^{q} x_{ij} \leq B x_{jj} \qquad \text{for all } i, j \tag{4.25}$$

and

$$\sum_{i=1}^{q} x_{ij} \geq 2 x_{jj} \qquad \text{for all } i, j \tag{4.26}$$

(III) Binary decision variables

$$x_{ij} = 0 \text{ or } 1 \tag{4.27}$$

A comparison between the models will follow later in the chapter.

4.2.1.1 Solution methodology

The improved models IMPGRP and NATGRP are still an integer programming formulation which can be solved by any procedure solving 0-1 LP model e.g. LINDO or CPLEX. Solution of the problem results into some non-zero variables which are interpreted for family formation.

Example problem

Take the example of Table 4.1 and use the distance matrix D.

The result we get in both the cases of natural grouping (NATGRP) and imposed grouping (IMPGRP) with required process families, p =2, are the same and are given below.

$x_{27} = 1$, $x_{77} = 1$, $x_{59} = 1$, $x_{99} = 1$, $x_{1109} = 1$ and all other x_{ij}'s $= 0$.

The process families are then {2,7} and {7,9,11} and part families are {1,3} and {2,4,5}. These results exactly match with the Kusiak's solution.

Some example problems (section 7.3) are presented in chapter 7 where the advantages of improved models IMPGRP and NATGRP can be clearly seen.

4.3 TYPES OF GENERALIZED GROUPING PROBLEMS

Problem Type I : Part-Wise Routeing Flexibility (PWRF)

This type of problem is generally of small order size (<10 parts).In this type, more than one type of operation of a part can be carried out either on a single machine or on more than one machine. This means that the set of alternative machines for a particular type of operation of the manufacturing system is different for different parts. This type of problem is generally of small order size (<10 parts). Example problem given in Table 4.1 is an example of this type of problem.

Problem type II : Operation-Wise Routeing Flexibility (OWRF)

This type of problem is for bigger manufacturing environment. In this type, alternative machines are available system-wise, that is the set of alternative machines for a particular type of operation is the same for all the parts over the whole manufacturing system. Example problem given in Table 7.7 is an example of this type of problem.

4.4 A HEURISTIC PROCEDURE

The proposed model, MAAM, (section 4.1) and the natural grouping model, NATGRP, (section 4.2.1) may result into a large number of process families for larger problem. So a heuristic method is applied for merging some of the process families which uses the same machines or the difference is only due to the alternative machines. Later the assignment of machines to the family is done while targeting the objective of minimizing intercell movements. A refinement procedure is also included which reassigns bottleneck parts to further reduce the intercell movement. The procedure consists of the following steps.

step 1 Identification of sets of alternative machines.

This step is useful only for generalized grouping problem of type II (OWRF).

When multiple routeing of part is possible because of presence of alternative machines for a type of operation. Identify those machines for different type of operations and make their sets. .

step 2 Merging the part families.

If the machines used by a process family is a subset or superset of the machines used by any other process family, merge the two process families. If subset or superset criteria fails between the two process families only due to the presence of machines of the same set of alternative machines then merge them by assigning either of the two machines. Restructure the process family accordingly. Repeat for every other pair of process family.

step 3 Assigning the machines to the process families.

(i) Compute for each machine m the usage factor u_{ml} for each process family l.

(ii) Assign the machine to the process family 'l' where u_{ml} is maximum, the ties are broken arbitrarily.

special case : Generalized grouping problem of type II (OWRF) will involve the following step also.

(iii) If $\sum u_{ml}$ for an alternative machine is zero i.e. no operation has been assigned to it, then place it in place of that machine of the same set which has $u_{ml} > 0$ for more than one process family. It may be noted that such a situation arises when number of alternative machines for a type of operation is relatively large.

step 4 Merging the process families and machine cells.

Merge two process families and their corresponding machine cells if it eliminates intercell movement and does not violate the cell size limit imposed. Stop when no merging is possible or the groups formed are disjoint.

step 5 Refinement procedure

Go to step 3. Assign the bottleneck parts to that cell which has maximum number of its operations. Ties are broken arbitrarily.

CENT LIBRARY
KANPUR
No. A. 122064

Note :

Simple grouping and generalized grouping type I problems need not to go through step1 and step (iii) step 3.

Kusiak and Agarwal's model need to go through steps 3 and 5.

4.5 COMPARISON OF MATHEMATICAL MODELS

A comparison of Kusiak's model (PMM), Agarwal's modified model, the proposed model (MAAM), and the proposed improvements in kusiak's model (IMPGRP and NATGRP) can be made on the basis of number of constraints, number of variables and the order of complexity. Order of complexity, as such, is calculated for the basic operations of an algorithm which are used repetitively. In that sense we do not have basic operations in an integer programming formulation as optimization is a *simultaneous* process. However, the number of all possible combinations which are likely to be explored while solving the problem through different integer programming formulations can be a good yardstick of measuring the complexity of the formulation. This has been discussed in the following paragraph.

The number of ways in which n different objects can be partitioned into p non-empty subsets is given by Stirling's number, $S(n,p) \cong p^n / p!$ (Duda and Hart, 1973). In most grouping problems, exploration of the total number of all feasible grouping alternatives for the optimal solution is simply not possible, even when large computers are available. The number of feasible clustering alternatives may be astronomically large even for relatively small grouping problems. For example, the total number of possible ways of grouping 20 objects in five groups is approximately 794,728,600,000. The combinatorial nature of the problem becomes more important when the number of groups is unknown, which is so in most grouping problems. In fact, this is what is to be determined in most of the problems. It can be seen that in this case the total number of possibilities is given by $\sum_{j=1}^{n} S(n,j)$. For the 20-object example, it would be of the

order of 10^{15}. A complete enumeration for the optimal solution should therefore be ruled out. This highly combinatorial nature of the grouping problem restricts the use of Kusiak's model to relatively small size problems. The proposed model applies the concept of associating one object with the other without having the number of clusters known. In that case the total enumeration of all possible combinations for n objects n(n-1). For 20-object example , this number would be 180. Thus, it can be concluded that the proposed model (MAAM) is highly computationally efficient even for the large scale problem. Similar discussion can be made for the natural grouping model (NATGRP) also.

Following tables shows comparison between the models explicitly:

Table 4.3 : Total number of enumerations for various models

Models	Total number of enumerations
Kusiak's model (PMM)	$\prod_{k=1}^{K} \lvert F_k \rvert \bullet S(K, p)$
Agarwal's modified model	$\prod_{k=1}^{K} \lvert F_k \rvert \bullet K c_p \bullet S(K-p, p)$
Proposed model (MAAM)	$\prod_{k=1}^{K} \lvert F_k \rvert \bullet K(K-1)$
Imposed grouping model (IMPGRP)	$\prod_{k=1}^{K} \lvert F_k \rvert \bullet K c_p \bullet S(K-p, p)$
Natural grouping model (NATGRP)	$\prod_{k=1}^{K} \lvert F_k \rvert \bullet K^2$

where

$\prod_{k=1}^{K} |F_k|$ = total number of ways selecting one process plan for each part k.

and ${}^{K}c_p$ = Total number of combination of selecting p elements out of K elements (K > P).

Table 4.4 compares the models on other criteria.

Table 4.4 : Comparison of various models

Criteria	Kusiak's model (PMM)	Agarwal's modified model	IMPGRP model	NATGRP model	Proposed model (MAAM)
GT concept	Violates	Violates	Violates	Agrees	Agrees
Kind of grouping	Imposed grouping	Imposed grouping	Imposed grouping	Natural grouping	Natural grouping
Number of variables	q^2	q^2	q^2	q^2	q^2
Number of constraints	q^2+K+1	q+K+1	2q+K+1	2q+K	2K+q
Heuristic application	steps 1,3 and 5	steps 1,3 and 5	steps 1,3 and 5	All steps	All steps

5

SIMILARITY AND DISTANCE MEASURES

In this chapter we discuss some similarity and distance measure developed along with some existing one from literature. These measures has been developed in the need of various types of inputs like the traditional binary input matrix, process time input matrix, operation sequence of process plans along with machine utilization penalty inputs.

We have observed for the various techniques that it is the measure of similarity (s_{ij}) or dissimilarity (d_{ij}) between the process plans which plays crucial role in optimization. So better or stronger is this measure, more practical the result will be. Also for different types of inputs e.g. binary input, process-time input, routeing sequence input and machine utilization input etc., one should have some measure which will be able to rightly account for the similarity or the dissimilarity between the process plans. We propose three types of coefficients for different types of inputs described in the following sections along with their parallel coefficients available in the literature.

5.1 MACHINE WEIGHTED SIMILARITY COEFFICIENT (MWSC)

This is defined for the binary matrix input. A part either requires a machine or does not, i.e.

a_{im} = 1, indicates plan i uses machine m;

= 0, otherwise.

The weightage to a machine is given by the sum of 1's under that machine in the incidence matrix. The significance of this is that the machine which is required by larger number of plans is more likely to be a general purpose machine on which a big investment has occurred. So, obviously the management would like to utilize it highly.

The weightage for a machine m w_m is thus given by

$$W_m = \sum_{i=1}^{N} a_{mi} \qquad (5.1)$$

for any two plans i and j

$$MWSC_{ij} = \sum_{i=1}^{N} w_m \cdot d(a_{im}, a_{jm})$$

$$= -\infty ;\ I, j \in F_k$$

$$= 0 ;\ \text{if } i = j \qquad (5.2)$$

where $\delta(a_{im}, a_{jm}) = 1 ;\ \text{if } a_{im} = a_{jm} = 1$

$$= 0 ;\ \text{otherwise}$$

5.2 PROCESS TIME RELATED COEFFICIENTS

Till now we considered 0-1 part machine matrix input to calculate the coefficient between two plans. Considering process time of operation as input, one should have some defined coefficient. In the following paragraphs we propose some coefficients based on the process time.

5.2.1 Multiplicative Weighted Similarity Coefficient (MuWSC)

Mosier (1985) defined three types of similarity coefficient between machines which use process time of operation as input. In parallel to his multiplicative weighted similarity coefficient between the pair of machines we propose a similarity coefficient between the pair of process plans.

The proposed coefficient is defined for operation process times as inputs. The similarity between a pair of process plans is the difference of the weights of all the common machines required by them and weights of the machines required by any one of the two process plans. The measure evaluates the similarity between the plans by taking products of the weight of the common machines and subtracting the weights of the machines required by any of the two plans.

Let t_{im} = total processing time of plan i on machine m ($t_{im} > 0$ only when $a_{im} = 1$).

For the computation of the proposed similarity coefficient $MuWSC_{ij}$, weight of machine is taken as the relative weightage on the basis of its requirements with respect to overall requirements of machines and is given as

$$W_m = \frac{\sum_{i=1}^{q} t_{im}}{NNON_{im}/M}$$

where $NNON_{im}$ = the total number of non zero elements in the matrix.

The numerator of W_m represents the total processing time on machine m over all plans. The denominator denotes the average time load over all machines.

The Multiplicative Weighed Similarity Coefficient between a pair of process plans is given by

$$MuWSC_{ij} = \frac{\sum_{m=1}^{M} t_{im} \cdot t_{jm} - \sum_{m=1}^{M} (t_{im} - t_{jm})^2}{\sum_{m=1}^{M} W_m^2} \qquad (5.4)$$

The first term in the numerator of the $MuWSC_{ij}$ is a measure of the requirement of the same machines by plans i and j. similarly the second term of the numerator is a measure of the misses of the plans with regard to machines. The term in the denominator is to standardize the coefficient in the range of [0, 1]. $MuWSC_{ij}$ can be used in place of s_{ij} in all the objective functions for maximization in the formulations of chapter 4.

5.2.2 Time of Movement Distance Coefficient (TMDC)

This is the sum of times of those operations of plan i and plan j which are not common.

Let the function $f(t_{im}, t_{jm}) = t_{jm}$; if $t_{im} = 0$

$= 0$; otherwise

This function gives the time of intercell movement when the machines required for the plan j is not available in plan i.

Let TOM_{max} = the maximum total time of movement when operations of plan j are carried out on the machines corresponding to plan i.

$$= \underset{i,j}{Max}\left[\sum_{m=1}^{M} f(a_{im}, a_{jm}) \right] \qquad (5.5)$$

The distance coefficient $TMDC_{ij}$, when the operations of plan j are carried out on the machines corresponding to plan i, is given by

$$TMDC_{ij} = \frac{\sum_{m=1}^{M} f(a_{im}, a_{jm})}{TOM_{max}} \qquad (5.6)$$

It can now easily be inferred from the above definition that

$$TMDC_{ij} \neq TMDC_{ji}$$

It is because machines requirement by the two plans i and j may differ.

Application of TMDC will result into assignment of machines to a process family to minimize the time of intercell traffic. $TMDC_{ij}$ can also be used in place of d_{ij} in all the objective function for the minimization in the formulations of chapter 4.

5.2.3 Weighted Part and Time Similarity Coefficient (WPTSC)

Application of MuWSC does ensure the minimization of intercell movement of parts as it tries to keep those parts together which have similarity of types of operations in terms of machine requirement. But this may not simultaneously ensure that the time of stay outside the cell due to this intercell movements are also a minimum. Hence, the TMDC should also be taken into consideration to have a tradeoff between the minimization of number of part movements and the minimization of time of movement. This is to be noted that we are only considering the time of operation while determining the time of movement not the time of transportation between the cells. Hence, another coefficient, to be called as Part and Time Similarity Coefficient (WPTSC), can be defined as follows.

$$WPTSC_{ij} = I_p * MuWSC_{ij} - I_t * TMDC_{ij} \qquad (5.7)$$

where I_p and I_t are the importance factors for part and time of movement, respectively.

Clearly, $WPTSC_{ij} \neq WPTSC_{ji}$.

5.3 OPERATION SEQUENCE BASED COEFFICIENTS

Operation sequence is considered to be one of the important criteria to judge the similarity between the process plans. This has an important influence on the cell layout and thereby on the material handling cost and time. The process plans having the highest similarity between them when grouped together will enhance the flowability of the components and will reduce the load on the problem of scheduling. Before presenting the proposed coefficient we discuss some of the earlier proposed coefficients grabbing the similarity between the process plans due to operation sequence.

5.3.1 Transformation Based Similarity Coefficients (TBSC) [Tam, 1990]

Tam (1990) described three types of transformations viz. substitution, insertion, and deletion to capture the operation sequence similarity between the process routeing of parts. The operation sequence of a plan is represented by a string having the indices of machines required by a part. The similarity coefficient defined by Tam specifies the similarity between the parts as the similarity between these strings in terms of the minimum number of weighted transformation required to carryout the conversion.

Definitions

Let the conversion of plan i to plan j require at least the following transformations:

n_s : number of interchanges(substitutions) in the string of plan j

n_i : number of insertions to the string of part j

n_d : number of deletion from the string of part j

Let M_1, M_2 and M_3 be the weights assigned to n_s, n_i and n_d respectively.

The similarity coefficient between plan i and plan j belonging to different parts is:

$$s_{ij} = 1/\min(M_1 n_s + M_2 n_i + M_3 n_d) \qquad (5.8)$$

5.3.2 Sequence Length Based Similarity Coefficient (SLBSC) [Choobineh, 1988]

Here it is assumed that similarity between part routeing is proportional to the length of common operations sequences between them. Higher the number of common operations required by components *in sequence*, higher is the similarity between them.

Definitions

The SLBSC is defined as the sum of similarity between the common operation sequences of length 1 to Le between any two parts.

Let

Le : the maximum length of sequence of operation considered for comparison

x_{oi} = 1, if operation o is performed on plan i;

= 0, otherwise.

$C_{ij}(y)$: number of common operation sequences of length y between plan i and plan j's routeings.

L_i : length of routeing of plan i

L_{min} : $\min_i (L_i)$ = number of operations in the smallest part routeing

The similarity is defined as

$$s_{ij} = \frac{1}{Le}\left[\frac{\sum_{0=1}^{M} x_{oi}\cdot x_{oj}}{\sum_{0=1}^{M}(x_{oi}+x_{oj}-x_{oi}\cdot x_{oj})} + \sum_{y=2}^{Le}\frac{C_{ij}(y)}{L_{min}-y+1}\right] \qquad (5.9)$$

The first term in the similarity coefficient defined is an another way of representing Jaccard similarity coefficient which is defined as the ratio of number of machines required by both plans to the sum of the machines required by both and the machines required by either of them. So this term represents the similarity of common operations between the plans. In effect, it represents the similarity due to single length of operation sequence. The second term calculates the similarity between the common operation sequences of the length 2 to L_e. This similarity coefficient is dependent upon the choice of L_e. Its high value while results in more accurate measurement of similarity coefficient, it involves more computational time. Generally its value is kept

equal to the maximum length of common sequence that exist between plans of two different parts.

5.3.3 Operation Sequence and Machine Utilization based Dissimilarity Coefficient (OSMUDC)

The proposed OSMUDC is as follows. A process plan has certain sequence of operation which is called process routing sequence. It is not necessary that two process plans which visit the same type of machines may have the same sequence of visiting them. The similarity coefficients discussed above do not consider the sequence possibility in the reverse direction which is likely, however, to avoid this we can put a high penalty factor on reverse flow of a part in the cell layout. In the case of SLBSC, the Jaccard coefficient does take the reverse flow into account in some sense but not the similarity of sequence in reverse direction. Sometime it is found that if a few back flow in the middle of flow of the component through the layout is allowed, then it might be possible that afterwards the flow is smooth in forward direction which is not accounted by TBSC and SLBSC. Hence, a more practical similarity from the manufacturing point of view can be assessed by the similarity of the sequence of operation in both forward as well as backward direction, however, to again avoid backward flow we penalize the back flow of part. To account the similarity due to similar types of machines used by two routeings, we allow reverse flow of component. The explanation of terms used for defining the dissimilarity coefficient is given below.

Definitions

Intercellular moves

Given the layout of machines of a machine cell, a part moves within the cell according to its requirement for operation in a certain sequence. These moves from machine to machine within the cell are termed as intracellular moves.

Intracellular moves

Given the layout of machines of a machine cell, a part may have certain requirement of machines for which the machine is not available in the cell and for that it has to visit some other machine cell. Every such move is termed as intercellular move.

Utilization penalty function

This is assigned to a machine by the management and depends on how and to what extent the management wants the machine to be utilized under different demand scenario for its products. A general purpose machine, on which higher investment is incurred should be maximum utilized. Similarly a special purpose machine should be aimed at to be idle for the least period. When the demand structure of products changes, a readjustment is supposed to be performed. The processing capability and flexibility of a machine should always be kept in mind while determining this. It is quite possible that the process plans also get redefined in such cases. In this sense we can say that this involves complete design of GT problem. It becomes a complex function when factors like cost, life, maintenance etc. are also accounted besides the above mentioned complexity of the environment. In general we can assume the utilization factor to be equal to the frequency of occurrences of the machine in the system. Whenever, two or more machines are desired to be kept together in one cell their utilization factor should be equally high.

5.3.3.1 Calculation of moves, penalty and dissimilarity coefficient

Notation

w_b = penalty factor for back flow ; $w_b \geq 1$.

w_i = weightage for one intracellular move.

w_o = weightage for one intercellular move.

E_i = set of machines required by plan i.

UP_m = utilization penalty for machine (m = 1 to M)

$TUP_{ij} = \sum UP_m$, $m \in E_i$ and $m \notin E_j$.

Table 5.1 : Intracell and intercell movements for plan j as per plan i

Machine for plan j	Intracell movements	Intercell movement	Remarks
1	4	no	forward flow
2	3	no	back flow
3	1	no	forward flow
4	1	no	- do -
5	-	yes	
8	3	no	forward flow
7	-	yes	

Thus the total intracell movements, $Xij = 4 + 3.w_b + 1 + 1 + 3$

Another way of counting intracell movement can be to count '0' movement for the forward movements and actual movements for the backward movements. That is,

$$Xij = 0 + 3 + 0 + 0 + 0 = 3.$$

It can be noted that in either case $X_{ij} \neq X_{ji}$.

While going through layout we also observe that locations of machines 5 and machine 7 of plan j are not found. So for those machines we count as number of intercell movement. That is the number of machines of plan j which is absent in plan i for the present case is $Y_{ij} = 2$.

Also, $Y_{ij} \neq Y_{ji}$

This also can be observed that machines 6 and 9 of plan i are not being utilized by the plan j so we charge utilization penalty for them. Here, the sum of the penalty of those machines of plan i which are not utilized by plan j, $TUP_{ij} = UP_6 + UP_9$. It can be easily seen that $TUP_{ji} \neq TUP_{ij}$.

The dissimilarity coefficient, d'_{ij}, when plan j proceeds in the machine layout of plan i

$$d'_{ij} = \frac{w_i \cdot X_{ij} + w_o \cdot Y_{ij}}{w_i \cdot X_{max} + w_o \cdot Y_{max}} + \frac{TUP_{ij}}{\sum_{m \in E_i} UP_m} \qquad (5.10)$$

Again, it can be noted that $d'_{ij} \neq d'_{ji}$

Hence the net dissimilarity between plan i and plan j is,

$$d_{ij} = d_{ji} = (d'_{ij} + d'_{ji})/2 \qquad (5.11)$$

From the final result of grouping we will have a sound indication of making the layout of machine for all process family. For making the final layout of machine cells we follow the following rule.

For a process family determine for its every plan i

$$D_i = \sum_{j \neq i} d'_{ij} \qquad (5.12)$$

where i and j belong to the given family.

The layout of machines for the process family will be according to that plan whose D_i is minimum of all.

6

MEASURES OF EVALUATION

Practical problems of group technology are usually and unfortunately framed according to the requirement of the methodologies. Many of the aspects of production (e.g. part machine incidence, their process time, operation sequence, multiple routeings, machine capacity constraint, production rate, demand, availability of material handling equipment, and many other such related to a manufacturing environment) are discarded. However, in some cases important ones are tried to get accounted for while developing a technique. Consideration of all those aspects which have an impact on the final solution will end up with a too complex problem to even formulate. So, at least some of them which play a greater role in decision making should be considered for the purpose of framing the problem. Hence, solution of practical problem without adequate generalization and theoretic formulation is supposed to be useful only in specific situation for which the problem has been framed. Such results can not usually be applied to other situation with different problem scenario. Group technology has suffered from this malady for a long time. When researchers and practitioners claim success with a certain method it is necessary to evaluate the result on an absolute quantitative scale. The practical use of such a measure is that subjective claim can be compared objectively.

The measures selected here evaluate the solutions of binary matrix, process time input matrix, and part routeing sequence. The measures to evaluate the binary matrix tends to compare the solution with perfect block diagonalized matrix and those for routeing sequence compare with a hypothetical solution with zero intercell traffic. For evaluating the solution of process time input, and the operation sequence input global evaluation criteria as suggested by Harhalakis et al.(1990 a) will be used.

Another measure for solution of process time input can be the number of parts movement along with the time of interaction among the cells.

6.1 EFFICIENCY MEASURES TO EVALUATE THE SOLUTION CONSIDERING MACHINE PART MACHINE MATRIX

The measures to compare the part machine matrix take a real value according to the row and column arrangement of part machine matrix. The output matrix provided by cell formation algorithm happens to be arranged in a way that all the non zero elements are concentrated towards the diagonal of the matrix. These efficiency measures evaluate the concentration of non zero elements (1's) in the matrix. The more 1's are arranged along the diagonals the better is the output depicted by the value of efficiency measures. The various measures incorporated in this work are grouping efficiency [Chandrasekharan and Rajagopalan, 1985], grouping efficacy [Kumar and Chandrasekharan, 1990] and relative efficiency [Chandrasekharan and Rajagopalan, 1987]. These are reproduced synoptically in the following subsections for the purpose of evaluation of various grouping approaches discussed in the Chapter 4.

6.1.1 Grouping Efficiency

The concept of grouping efficiency is developed to provide a quantitative standard on a rational scale for comparing different solutions by different approaches to the same problem.

Let $A = [a_{ij}]$ be the part-machine binary (0-1) incidence matrix, i = 1,.....,N and j = 1,.....M.

Let $\{D_r\}$, where r = 1,2.....R, be the diagonal submatrices obtained by associating the product group with the machine group in the diagonalized matrix (R = total number of such groups). A perfect grouping implies that

$a_{ij} = 1$ if $a_{ij} \in \{D_r\}$ and

$= 0$ if $a_{ij} \notin \{D_r\}$ r = 1.....R

The diametrically opposite situation, therefore implies that

$a_{ij} = 0$ if $a_{ij} \in \{D_r\}$ and

$= 1$ if $a_{ij} \notin \{D_r\}$ $r = 1.....R$

The efficiency scale is defined to cover these extreme cases, its value being set at one for the former case and zero for the later.

Goodness of grouping depends on two aspects: within group utilization and intercell movement. From the matrix point of view the concentration of non zero elements in the diagonal submatrix refers to utilization and the presence of such elements outside the diagonal submatrices represents intercell movements. A better grouping therefore means an increase in the utilization, a decrease in the intercell movement or both. The number of non zero elements in the diagonal blocks is given by

$$e_d = \sum_{r=1}^{R} \sum_{i=M_{r-1}+1}^{M_r} \sum_{j=N_{r-1}+1}^{N_r} a_{ij}$$

$$(M_0 = 0, N_0 = 0, M_R = N, N_R = M) \qquad (6.1)$$

N_r = Sum of number of parts cluster 1 through cluster r.

M_r = Sum of number of machines cluster 1 through cluster r.

and the number of non zero elements outside the diagonal block is

$$e_o = e - e_d$$

Where 'e' is the total number of non-zero elements (1's) in the matrix.

and the number of voids in the diagonal blocks is

$$e_v = \sum_{r=1}^{R} M_r N_r - e_d$$

Grouping efficiency can therefore be considered as a weighted average of two efficiencies η_1 and η_2 defined as follows.

$$\eta_1 = \frac{e_d}{\left(\sum_{r=1}^{R} M_r N_r\right)} \qquad (6.2)$$

$$\eta_2 = 1 - \frac{e_o}{(MN - \sum_{r=1}^{R} M_r N_r)} \qquad (6.3)$$

The numerator of the expression for $\eta 1$ is the number of non zero elements in the diagonal block and the denominator the total number of elements there in. Similarly $\eta 2$ is the ratio of the number of zeros in the off diagonal block to the total number of elements therein.

The grouping efficiency 'η' can be expressed as the weighted average of η_1 and η_2

$$\eta = q\eta_1 + (1\text{-}q)\eta_2 \qquad (6.4)$$

η satisfies the basic requirements of non dimensionally, non - negativity and the range of 0 to 1. The weighing factor enables the analyst to alter the emphasis between utilization and intercell moves. A high value of q ensures greater utilization.

Limitations of grouping efficiency

1. *Effects of weights*

The weighted average of η_1 and η_2 gives the analyst enough freedom to decide the relative importance between intercell movements and voids in diagonal blocks. Chandrasekharan and Rajagopalan [1985] concluded that a value of q = 0.5 leads to the situation of equal weights to voids and exceptional elements. An analysis of the expression would reveal that this is not true. When large matrices are diagonalized, for any number of blocks greater than two it can be observed that the first term in expression has a much smaller denominator compared to the second term, whereas the numerator are more or less of the same order. As the matrix size increase this disparity between the first and the second term widens and for large and / or sparse matrices the second term becomes less and less effective. In fact if q and 1-q are of the same order, the effect of intercell movement is never reflected in the efficiency value in the case of

large and sparse matrices. Therefore for equal weight to voids and exceptional elements it is necessary to choose low value q. Further it is desirable to bring the two parts of the expression under a common denominator.

2. *Effect of sparseness of matrix*

Choice of q would be more rational if it is linked with the size and sparsity of the matrix. As a result, q is chosen as the density of an imaginary matrix that has an identical and perfect block diagonal structure as the given matrix.

$$q = \sum_{r=1}^{K} M_r N_r / MN \qquad (6.5)$$

The efficiency value reduces to

$$\eta = 1 - (e_V + e_0) / MN \qquad (6.6)$$

This simplified expression assumes that the perfect diagonal form has 100 percent efficiency which is reduced by the presence of voids and exceptional elements.

The following characteristics are important in the case of the modified expression for η after choosing q as discussed earlier.

The main difference between this expression and the earlier expression for efficiency is that the present expression has the size of the matrix (MN) as its denominator, whereas in earlier expression $\eta 1$ and $\eta 2$ have two different denominators. It is also interesting to note that it is possible to assign relative weights for voids and exceptional elements, because it is in the present form that voids and exceptional elements have equal weights.

A major defect of grouping efficiency in both forms is its low discriminating power. It is obvious that $(e_V + e_0)$ is much less than the size of the matrix (MN). Thus for large matrices the term $(e_V + e_0) / MN$ will take values of the order of 0.001 which leads to an efficiency close to one.

Another serious defect lies in the definition of zero efficiency. Zero has been defined as the most degenerate case where $(e_V + e_0) = MN$. This implies that the diagonal blocks are full of zeroes and off diagonal and off diagonal are full of ones. It can be seen that this definition is realistic from physical point of view.

3. *Effect of the size of the matrix*

By accepting 'e' as the denominator of the efficiency function it will have the form

$$\eta = 1 - [(e_V + e_0)/e]$$
$$= (e_d - e_V)/e$$

The main drawbacks of this function are as follows :

1. Zero point of the efficiency value is not properly defined. The value of η becomes 0 when $e_d = e_V$ i.e.

   $$e_d = e_V = 1/2[(\sum_{r=1}^{R} M_r N_r)]$$

   This means that zero efficiency is dictated by the fact that there are 50 percent voids in the diagonal blocks irrespective of the number of exceptional elements present.

2. A major defect of grouping efficiency (both in earlier and modified form) is its discriminating power. It is obvious that $(e_d + e_V)$ is much less than the size of the matrix. Thus for a large matrices the term $(e_V + e_0)/MN$ will take values of the order of 0.001 which leads to an efficiency close to one.

3. This function takes negative values when $e_V < e_d$.

6.1.2 Grouping Efficacy

For a large denominator the discriminating power of the efficiency function is poor and for a small denominator the function may itself become negative.

The meaningful denominator between 'e' and 'MN' is the operational zone of the matrix. The operational zone of block diagonal form consists of the diagonal blocks usually consists of both ones and voids, and the exceptional elements consists of only ones.

Operational zone = (Voids + ones) in the diagonal blocks + ones in off diagonal blocks.

$$= e_V + e_d + e_0$$
$$= e_V + e$$

Thus expression can be modified as

$$\eta = 1 - [(e_v + e_o) / (e_v + e)]$$

$$= (e - e_o) / (e_v + e)$$

This is the grouping efficacy (GEF).

$$GEF = (1 - \varphi) / (1 + \phi) \qquad (6.7)$$

φ = Number of exceptional elements / total number of operations

ϕ = Number of voids in diagonal blocks / total number of operations

6.1.3 Relative Efficiency

The *limiting efficiency* states that for any block diagonal structure defined by sequence of row clusters and column clusters, there exists a limit on the maximum attainable efficiency.

Let M_r and N_r be the number (dimension) of vectors in the rth row cluster and column cluster respectively.

Arrange the sequence of integers 'N' and 'M' so that

$$M_1 \leq M_2 \leq \ldots\ldots M_R$$

and

$$N_1 \leq N_2 \leq \ldots\ldots N_R$$

Then accommodation 'S' in the diagonal block is maximum when M and N are correlated to form a diagonal submatrix. Then

$$S_{max} = \sum_{r=1}^{R} M_r N_r$$

The accommodation 'S' is minimum when sequence of 'M' and 'N' are reversed and the corresponding cluster M and N are correlated to form diagonal blocks. Then

$$S_{min} = \sum_{r=1}^{R} M_r N_r$$

for any general combination

$$S_{min} \leq S \leq S_{max}$$

The grouping efficiency (Chandrasekharan and Rajagopalan, 1985) is given by

$$\eta = q\eta_1 + (1-q)\eta_2$$

Case1 : ($e<S_{min}$)

In this case voids will appear in blocks, but exceptional elements can be hypothetically avoided. Hence the limiting value of efficiency suffers because of first component of grouping efficiency η_1.

Case2 : ($e>S_{max}$)

In this case exceptional elements are inevitable, although voids can be theoretically avoided. Hence the second component of grouping efficiency η_2 suffers.

$$\eta_o = q + [(1-q)*(MN - e/MN - S_{max})]$$

Case3 : ($S_{min} \leq S \leq S_{max}$)

For any value of 'S' between S_{min} and S_{max} it is theoretically possible to find a sequence so that

$$S \approx e$$

In this case, $\eta_o = 1$.

Relative efficiency is defined as the ratio of grouping efficiency and limiting efficiency. This ratio is a useful check to find out whether a given algorithm has achieved the best of what is achievable under the constraints of given data.

6.2 GLOBAL EFFICIENCY MEASURES

This class of measures evaluate the result which are either in the form of binary matrix, process time matrix or in the form of part routing sequence. Three measures which have been designed to evaluate their results, by Harhalakis et. al.(1990a) are as follows.

6.2.1 Global Efficiency

This is the ratio of the total number of part operation that are performed within their respective cells to the total number of operations in the system.

$$\text{Global efficiency} = \frac{\sum_{i=1}^{N} s_i}{\sum_{i=1}^{N} r_i} \qquad (6.8)$$

where s_i is the number of operations in the r_i that are performed in the cell corresponding to part i.

Global efficiency examines the grouping power of any method in a absolute way. It reflects the effectiveness of the assignment in confining the operations of parts within their respective cells. It assigns a penalty for each operation to be performed on a part outside the cell.

6.2.2 Group Efficiency

This criterion evaluate the efficiency of the solution to minimize the intercell moves. The intercell moves are maximum, when every machine belongs to a single group and are zero when all machines are in a single group.

The ratio of the difference between the total number of maximum external cells that could be visited and the total number of external cells actually visited by all parts to the total number of maximum cells that could be visited by them is known as group efficiency.

The maximum number of external cells that could be visited is

$$E_W = \sum_{i=1}^{N} \text{Min} \{ (q_i - 1),(w-1) \} \qquad (6.9)$$

where q_j is the number of different machines belonging to the same routing r_j

X_{ic} = 1 if part i visits cell c

= 0 otherwise $\qquad c = 1.....w$

The total number of cells actually visited by all parts,

$$A_W = \sum_{i=1}^{N} \sum_{c=1}^{w} (X_{ic} - 1) \qquad (6.10)$$

$$\text{Group efficiency} = \frac{E_W - A_W}{E_W} \qquad (6.11)$$

This ratio assigns the same penalty whether there is one or more operations that are performed on the part in the same foreign cell. This is a more abstract measure, but it

reflects the effectiveness of the assignment to confine the parts to as few foreign cells as possible.

In this measure of evaluation the penalty given account for the number of cells in which the external operations are performed. Global efficiency gives the penalty regardless of the number of external cells visited. So the higher the group efficiency, the better the assignment is. There must be at least two cells in the system for the group efficiency to be calculated.

6.6 Group Technology Efficiency

It is the ratio of the difference between the maximum number of intercell traffics possible and the number of intercell traffics actually being required by the system to the maximum number of intercell traffics possible.

The maximum number of intercell traffics possible in the system is

$$I = \sum_{i=1}^{N} (r_i - 1) \qquad (6.12)$$

$X_{io} = 0$, if operations o and o+1 are performed in the same cell;

$= 1$, otherwise.

where $o = 1 \ldots\ldots\ldots r_i - 1$, for part i.

The number of intercell traffic actually required by the system is

$$U = \sum_{i=1}^{N} \sum_{o=1}^{r_i - 1} X_{io} \qquad (6.13)$$

Hence, Group technology efficiency $= (I - U) / I$ (6.14)

This is a very powerful criterion, which takes into account the sequence in which the operations are performed apart from the cell in which they are performed. This efficiency provides a penalty of 1 for each intercell move. Alternatively, this efficiency can be evaluated by using the actual production volume of the parts (u_j) while computing the intercell moves, I and U, thus providing a weighed penalty. it can be viewed as a measure reflecting the saving in intercell moves.

7

IMPLEMENTATION, EXPERIMENTATION, RESULTS, AND OBSERVATIONS

This chapter presents its contents in three phases.

Phase I : Experimentation and comparison of the proposed model (MAAM) with Kusiak's and Agarwal's modified models.

Phase II : Validation of the proposed similarity and dissimilarity coefficients.

Phase III : Testing of the proposed improvements in Kusiak's model (PMM).

To carry out three-phase experimentation, an integrated system is developed and implemented on HP 9000 machines in C language. The system gives mathematical formulation for any combination of the model and similarity/dissimilarity coefficient desired. All the five models (Kusiak's PMM, Agrawal's modified model, MAAM, IMPGRP and NATGRP) have been included. Four coefficients viz. Lee's dissimilarity coefficient, machine weighted similarity coefficient (MWSC), weighted part and time similarity coefficient (WPTSC), and the operation sequence machine utilization dissimilarity coefficient (OSMUDC) are incorporated in the system. Thus, in total 5x4 = 20 different types of possible formulations are experimented with. The system is menu driven in which user can select the combination of the model and the coefficient to be used to formulate the problem. The formulation thus resulting is stored in an output file which is solved by CPLEX. Computational CPU times are recorded. The output from CPLEX is interpreted to form the process families. The proposed heuristic

is then applied manually to merge the process families and then to form part families and machine groups. Various grouping efficiency measures for the final solution are computed manually.

- Phase I makes use of the integrated system for formulating some example problems using the proposed model (MAAM), Kusiak's model (PMM) and Agarwal's modified model. These models are then compared on the criteria of CPU-time performance and efficiency measures.
- Phase II validates the proposed similarity/dissimilarity coefficients by testing some example problems containing the inputs required for the calculation of the coefficients. The testing makes use of the system to apply either the proposed model (MAAM) or Kusiak's model (PMM) for solving the problem.
- Phase III makes use of the system for testing some example problems on IMPGRP and NATGRP models. The results are then compared with those obtained by applying original Kusiak's model (PMM). The basis of comparison is again CPU-time and efficiency measures discussed in Chapter 6. Some noteworthy observations are also discussed.

PHASE I

7.1 COMPARISON OF PROPOSED MODEL (MAAM) WITH KUSIAK'S AND AGARWAL'S MODIFIED MODELS

The example problems taken are binary (0-1) plan and/or part machine incidence matrix. The problems are divided into two categories: (a) simple grouping problem with each part having single process plan, and (b) generalized groping problem with parts having alternate process plans.

7.1.1 Simple Grouping Problem

The 0-1 part machine incidence matrix for five simple grouping example problems are shown in Tables 7.1 to 7.5.

Example Problem 1

Table 7.1 : Part-Machine Incidence Matrix for Example Problem 1

		Machines									
		1	2	3	4	5	6	7	8	9	10
	1	0	0	1	1	0	1	0	0	0	0
	2	1	0	0	0	0	0	1	0	0	1
	3	0	1	0	0	1	0	0	1	0	0
	4	0	0	0	1	0	1	0	0	1	0
	5	0	1	0	0	1	0	0	1	0	0
P	6	0	0	1	0	0	1	0	0	1	0
a	7	0	0	0	0	0	0	1	0	0	1
r	8	0	1	0	0	1	0	0	1	0	0
t	9	0	0	1	1	0	1	0	0	1	0
s	10	1	0	0	0	0	0	1	0	0	1
	11	1	0	0	0	0	0	1	0	0	1
	12	1	0	0	0	0	0	1	0	0	1
	13	0	1	0	0	1	0	0	1	0	0
	14	0	0	1	1	0	1	0	0	1	0
	15	0	1	0	0	1	0	0	1	0	0

Example Problem 2

Table 7.2 : Part-Machine Incidence Matrix for Example Problem 2

		Machines														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0
	2	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1
P	3	1	0	1	0	0	1	0	0	0	0	0	0	0	1	0
a	4	1	0	1	1	0	1	0	0	0	0	0	0	0	1	0
r	5	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1
t	6	0	1	0	0	0	1	1	0	0	0	1	1	0	0	0
s	7	0	1	0	0	0	1	0	0	0	0	1	1	0	0	0
	8	1	0	1	1	0	1	0	0	0	0	1	0	0	1	0
	9	0	0	1	1	0	1	1	0	0	0	0	0	0	1	0
	10	0	0	0	0	0	0	0	1	1	1	0	0	1	0	1

Example Problem 3

Table 7.3 : Part-Machine Incidence Matrix for Example Problem 3

		Machines										
		1	2	3	4	5	6	7	8	9	10	11
	1	1	0	0	1	1	0	0	0	0	1	0
	2	1	0	0	1	1	0	0	0	0	1	0
	3	1	0	0	1	1	0	0	0	0	1	0
	4	0	0	0	0	0	0	1	1	1	0	1
	5	0	1	1	0	0	1	0	1	0	1	0
	6	0	0	1	0	0	0	1	0	0	0	0
	7	0	0	0	0	1	0	0	0	0	1	0
	8	0	1	0	0	0	1	0	0	0	1	0
P	9	0	0	0	0	0	0	1	1	1	0	1
a	10	0	0	0	0	0	0	0	1	0	0	1
r	11	1	1	0	1	0	0	0	0	0	1	0
t	12	0	1	1	0	0	1	0	1	0	1	0
s	13	0	0	1	0	0	0	0	1	0	0	0
	14	0	0	0	0	0	0	1	1	1	0	0
	15	1	0	0	1	1	0	0	0	0	1	0
	16	1	0	0	1	1	0	0	0	0	1	0
	17	0	0	0	0	0	0	1	1	1	0	1
	18	0	0	0	0	0	0	1	1	1	1	1
	19	0	1	1	0	0	1	0	1	0	0	0
	20	1	0	0	1	1	0	0	0	0	0	0
	21	1	0	0	1	1	0	0	0	0	0	0
	22	1	0	0	1	1	0	0	0	0	0	0

Example Problem 4

Table 7.4 : Part-Machine Incidence Matrix for Example Problem 4

		Machines													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0
	2	0	0	0	1	0	0	1	0	0	1	0	0	0	0
	3	0	1	1	0	0	0	0	0	0	0	1	0	0	0
	4	0	1	1	0	0	0	0	0	0	0	1	0	0	0
	5	0	0	0	0	0	0	0	1	1	0	0	0	0	0
	6	1	0	0	0	0	0	1	0	0	0	0	0	1	0
	7	1	0	0	0	0	0	0	0	0	0	0	1	1	0
	8	0	0	0	0	0	0	0	0	0	0	0	1	1	0
P	9	0	0	0	0	0	1	0	1	1	0	0	0	0	1
a	10	0	0	0	0	0	1	0	1	1	0	0	0	0	0
r	11	0	0	0	0	0	1	0	0	0	0	0	0	0	1
t	12	0	0	0	0	0	1	0	1	1	0	0	0	0	0
s	13	0	0	0	0	0	0	0	0	1	0	0	0	0	1
	14	0	0	0	0	0	1	0	1	0	0	0	0	0	0
	15	0	0	0	0	0	1	0	1	1	0	0	0	0	1
	16	0	0	0	0	0	1	0	1	0	0	0	0	0	0
	17	0	0	0	1	1	0	1	0	0	0	0	0	0	0
	18	0	0	0	0	0	0	0	0	0	0	0	0	1	0
	19	0	0	0	1	0	0	1	0	0	0	0	0	0	0
	20	0	0	0	1	1	0	1	0	0	0	0	0	0	0
	21	0	0	1	0	0	0	0	0	0	0	1	0	0	0
	22	0	0	0	0	0	1	0	1	0	0	0	0	0	0
	23	0	0	0	1	1	0	0	0	0	0	0	0	1	0
	24	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Example Problem 5

Table 7.5 : Part-Machine Incidence Matrix for Example Problem 5

		Machines															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0
	2	0	1	0	0	0	1	0	1	1	0	0	0	0	1	0	1
	3	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0
	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	5	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
	6	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
	7	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1
	8	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0
	9	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0
	10	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1
	11	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
	12	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0
	13	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0
	14	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
	16	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	17	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0
	18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	19	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0
P	20	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
a	21	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0
r	22	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
t	23	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0
s	24	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0
	25	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
	26	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	27	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0
	28	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0
	29	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	30	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
	31	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
	32	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1
	33	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
	34	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
	35	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
	36	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	37	1	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1
	38	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1
	39	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
	40	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0
	41	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
	42	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1
	43	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0

7.1.1.1 Methodology, solution and observations

We find the solution of above problems using Kusiak's PMM model, Agarwal's modified model and the proposed model (MAAM). The coefficients used are: (a) Lee's distance coefficient, and (b) machine weighted similarity coefficient(MWSC).

For the proposed model number of part families to be formed is not prespecified. The proposed heuristic procedure is applied (excluding step1, which is applicable with generalized grouping).

For Kusiak's and Agarwal's modified models the number of maximum number of part families (or cells) is assumed and solution gives the part families formed. Hence, to reach at the best solution possible we run the model for a few times (trial and error). Step 3 of the heuristic proposed is used for machine assignments to part families. The refinement procedure of the heuristic improves the solution whenever it is possible to improve.

Solutions of Example Problem 1

Problem size : (15x10), total number of operations (e) = 46.

Solutions are shown in the subsequent tables (Table 7.6 and 7.7).

(a) Lee's distance coefficient

Table 7.6 : Solutions of various models using Lee's distance coefficient for Example problem 1

Model	p or number of loops l	CPU time (sec)	Part families	Machine cells	Number of bottleneck operations (e_o)	Number of voids (e_v)	Remarks
Kusiak's model (PMM)	p = 4	0.06	(1,4,6, 9,14) (2,10, 11,12) (7) (3,5, 8,13,15)	(3,4,6,9) (1,7,10) (-) (2,5,8)	2	3	Family of a single plan is formed.
	p = 3	0.05	(1,4,6, 9,14) (2,7,10, 11, 12) (3,5,8, 13, 15)	(3,4,6,9) (1,7,10) (2,5,8)	0	4	No single plan family.
Agarwal's modified model	p = 4	3.06	(1) (4,6,9,14) (2,7,10, 11 12) (3,5,8, 13, 15)	(-) (3,4,6,9) (1,7,10) (2,5,8)	3	3	Single plan family is formed.
	p = 3	1.55	(1,4,6, 9,14) (2,7,10, 11, 12) (3,5,8, 13, 15)	(3,4,6,9) (1,7,10) (2,5,8)	0	4	No single plan family.
Proposed model (MAAM)	l = 6	0.01	- do -	- do -	0	4	Cell size limit = 5

Observations

1. There is remarkable difference of computational time of three formulations. The proposed model (MAAM) takes the least time (0.01 sec.) and Agarwal's model takes the maximum time 3.06 sec. with four families (p = 4) and 1.55 sec. with 3 families (p = 3).

2. can observe the GT concept violation by both Kusiak's and Agarwal's modified models with p = 4 as single plan families (7) and (1) respectively are formed.
3. The sum total of the number of bottleneck operations (e_o) and the number of voids (e_v), that is, ($e_o + e_v$) is 6 for Agarwal's model (p = 4), 5 for Kusiak's model (p = 4), and 4 for the remaining cases of the Table 7.1. Based on ($e_o + e_v$) the solutions obtained by Kusiak's and Agarwal's modified models both with p = 4 is worst as ($e_o + e_v$) is high.
4. Bottleneck operations are more critical than void elements from the point of view of practical result. Hence, based on the number of critical operations (e_o), the solution obtained by Kusiak's and Agarwal's modified models, both with p = 4 are worst as e_o = 2 and 3 respectively for them, whereas for the proposed model (MAAM) there is no bottleneck operation.
5. Thus, in the case of Kusiak's and Agarwal's modified models with higher value of (say p = 4), while the values of ($e_o + e_v$) are higher i.e. the grouping quality is bad, simultaneously the families of single process plan also get formed and computational time also increases.
6. Only merging step of the proposed heuristic is required to get the final solution of the proposed model (MAAM).
7. The solution by the proposed model exactly matches with the best solution obtained by Kusiak's and Agarwal's modified models both for p = 3.
8. It can be argued that the total CPU time taken by the proposed model and heuristic together will still be lower because already the problem is reduced to loops which is at least half of the number of parts and the merging step have the order of complexity as $O(l^2)$.

Efficiency measures for the best solutions obtained by Kusiak's and Agarwal's modified models both with p = 3, and the proposed MAAM model are:

grouping efficiency	97.33%
grouping efficacy	80.76%
global efficiency	100.00%
group efficiency	100.00%

(b) MWSC

By using MWSC in the three models we achieve the best result ($e_o + e_v$) as obtained for p = 3 in Kusiak's and Agarwal's modified models and the proposed model in (a). Even using higher value of p (say p = 4) in Kusiak's and Agarwal's modified models, solution results into forming three part families. The CPU time is, however, different as shown in Table 7.7.

Table 7.7 : Computational times of various models using MWSC for example problem 1

Model	p or l	CPU time (sec)
Kusiak's model	p = 4	0.03
	p = 3	0.05
Agarwal's modified model	p = 4	36.22
	p = 3	20.30
Proposed model	l = 4	0.01

Observations

1. In comparison to (a) lesser CPU time for Kusiak's model and higher time for Agarwal's modified model are needed to solve the problem.
2. For Kusiak's and Agarwal's models both, no family of single process plan is formed with even higher value of p (say p = 4 or 5) and solution results into such a number of process families (less than p) which gives the best result possible (i.e. with p =3).

3. Because of this, CPU time for Kusiak's model with higher value of p (say p = 4) is less than that required for the best result possible i.e. with p = 3.
4. Less number of loops are formed for the proposed model. It is because of the somewhat uniform distribution of operations on machines in the example problem. That is most of the machine's weighing factor lies in a small range. For the large range case, i.e. large variability of operations assignments on machines, larger number of loops are expected.
5. Why to use MWSC ?

- Generally it is found that intercell moves are more due to the machine having more number of operations, so the aim is to keep those parts together creating these moves due to these machines.

6. Where to not use MWSC?

- When a problem has its total operations on machines distributed in a large range. That is one machine has few operations while some other machine has large number of operation assigned.

Solutions of Example Problem 2

Problem size : (10x15) and total number of operations (e) = 48.

Solution are shown in the Tables 7.8 and 7.9.

(a) Lee's distance coefficient

Table 7.8 : Solutions of various models using Lee's distance for example problem 2

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 3	0.02	(1,6,7)	(2,6,7, 11, 12)	1	10	No single family plan
			(2,5,10)	(5,8,9,10, 13,15)			
			(3,5,8, 13, 15)	(1,3,4,6, 7,14)			
Agarwal's modified model	p = 3	0.2	- do -	-do -	- do -	- do -	- do -
Proposed model (MAAM)	l = 3	0.02	- do -	- do -	- do -	- do -	- do -

Efficiency measures are:

grouping efficiency	92.66%
grouping efficacy	81.03%
global efficiency	97.91%
group efficiency	95.00%

(b) MWSC

We get the same results obtained in Table 7.8 but the computational times are different given in Table 7.9.

Table 7.9 : Computational time of various models using MWSC for example problem 2

Model	p or l	CPU time (sec)
Kusiak's model (PMM)	p = 4	0.03
	p = 3	0.02
Agarwal's modified model	p = 4	1.98
	p = 3	0.86
Proposed Model (MAAM)	l = 3	0.01

Solutions of Example Problem 3

Problem size : (22x11) with total number of operations (e) = 78.

Solutions are shown in the following tables (Table 7.10 and 7.11)

(a) Lee's distance coefficient

Table 7.10 : Solutions of various models using Lee's distance coefficient for example problem 3

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 3	0.29	(1,2,3,7,11, 15,16, 20,21,22)	(1,4,5,10)	15	10	Refinement procedure could not improve e_v with p = 3 because the procedure does not consider the void minimization which can, however, be added. for p = 5 merging step and refinement step reduces e_v and e_o both.
	or	0.16	(46,9,10, 14,17,18)	(7,8,9,11)			
	p =5		(5,8,12, 13, 19)	(2,3,6)			
Agarwal's modified model	p = 3 or p = 5	<= 30	- do -	- do -	- do -	- do -	- dc -
Proposed model (MAAM)	l = 9	0.03	(1,2,3,7,11, 15,16,20, 21,22) (4,9,10,14, 17,18) (5,6,8,12, 13,19)	- do -	14	10	e_v is reduced by applying refinement procedure. Cell size limit = 5

(b) MWSC

Table 7.11 : Solutions of various models using MWSC for example problem 3

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 3	0.15	(1,2,3,7,11, 15,16,20, 21,22)	(1,4,5,10)	15	10	Refinement procedure could not improve result as it does not consider void minimization which can be added.
	or						
	p = 5	0.13	(46,9,10, 14,17,18) (5,8,12, 13, 19)	(7,8,9,11) (2,3,6,)			
Agarwal's modified model	p = 3 or p = 4	<= 30	- do -	- do -	- do -	-do-	- do -
Proposed model (MAAM)	l = 8	0.03	(1,2,3,7,11, 15,16,20, 21,22) (4,9,10,14, 17,18) (5,6,8,12, 13,19)	- do -	14	10	e_v is reduced by applying the refinement procedure. Cell size limit = 5

Efficiency measures for (a) and (b) both are :

Efficiency measures	Kusiak's and modified Kusiak's model	Proposed model (MAAM)
grouping efficiency	89.67%	90.08%
grouping efficacy	73.11%	73.91%
global efficiency	87.18%	87.18%
group efficiency	73.68%	73.68%

Observations

In all the above three problem it is observed that

1. In most of the cases, CPU times with MWSC are found to be less than their corresponding cases of Lee's distance. In case of the proposed model less number of loops are formed for MWSC and simultaneously CPU time is also less which is due to somewhat uniform distribution of operations on various machines.
2. The proposed model takes minimum computational time as well as the final solution ($e_o = 14$, $e_o + e_v = 24$) is better than that by Kusiak's and Agarwal's modified models ($e_o = 15$, $e_o + e_v = 25$).
3. In Agarwal's modified model we had either large computational time or had to limit this.

Solutions of Example Problem 4

Problem size : (24x14), total number of operations (e) = 60.

Solutions are shown in Tables 7.12 and 7.13.

(a) Lee's distance coefficient

Table 7.12 : Solutions of various models using Lee's distance for example problem 4

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 4	0.38	(1,2,17,19, 20,23) (3,4,6,7,8,18, 21,24) (6,7,8,18) (5,9,10,11,12,13, 14,15,16,22)	(4,5,7,10) (2,3,11) (1,12,13) (6,8,9,14)	29	3	Refinement does not improve e_v and e_o further.
Agarwal's modified model	p = 4	<= 50	- do -	- do -	29	3	with refinement procedure otherwise $e_v = 30$, $e_o = 4$.
Proposed model (MAAM)	l = 9	0.07	- do -	- do -	- do -	- do -	Refinement procedure reduces e_v further. Cell size limit =5

Efficiency measures are:

Efficiency measures	All models
grouping efficiency	90.47%
grouping efficacy	64.44%
global efficiency	95.08%
group efficiency	91.89%

It was observed that Kusiak's model (p =3) results into three families (with $e_o = 53$ and $e_v = 3$) which is worse than the result with p = 4 in Kusiak's model (with $e_o = 29$ and $e_v = 3$, see Table 7.12).

(b) MWSC

Table 7.13 : Solutions of various models using MWSC for example problem 4

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 4	0.23	(1,2,17,19, 20,23) (3,4,6,7,8, 18,21,24) (5,9,10,11, 12,13,14, 15,16,22)	(4,5,7,10) (1,2,3,11, 12,13) (6,8,9,14)	53	3	After refinement procedure

Agarwal's modified model	p = 4	<= 50	same as (a)	same as(a)	29	3	No improvement by refinement procedure
Proposed model (MAAM)	l = 11	0.05	- do -	- do -	- do -	- do -	Refinement procedure reduces e_v and e_o both. Cell size limit = 5

Observations

In this problem also, use of MWSC with these models results in less CPU time. However larger number of loops results for the proposed MAAM model. Number of loops resulting from the proposed model is a direct consequence of the way the distribution of operations on machines. Larger number of loops results when the range of distribution of operations is large or the distribution is skewed, that is, the difference between minimum number of operation and the maximum number of operations assigned to the machines in the system is large. The measure of large variability of operations distribution on various machines can be determined by the difference between the number of operations assigned on various machines and the average number of operations on a machine in the system, which is the ratio of the total number of operations (e) and the total number of machines (M).

Efficiency measures are

Efficiency measures	Proposed and Modified model	Kusiak's model
grouping efficiency	90.47%	83.33%
grouping efficacy	64.44%	50.42%
global efficiency	95.08%	95.08%
group efficiency	91.89%	91.89%

In *one go*, Only the proposed model (MAAM) and the proposed heuristic together gives the optimal number of part families for which the efficiency measures are the best

possible. Kusiak's model has to be used few times to reach at this result. This can be validated by the reduction of efficiency measures with p = 3 and using Kusiak's model (see above table).

Solutions of Example Problem 5

Problem size : (43x16), total number of operations (e) = 126.

Solutions are shown in the subsequent tables (Table 7.14 and 7.15).

(a) Lee's distance coefficient

Table 7.14 : Solutions of various models using Lee's distance for example problem 5

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 5	3.30	(1,6,7,12,13,17,25,26, 31,34,35,36,39) (2,4,10,18,28,32, 37,38,40,42) (3,11,20,24,27,30) (5,15,16,22, 29,33,41,43) (8,9,14,19,21,23)	(3,6,7,10,14) (1,2,9,6) (8,11,12,13) (5,15) (4)	39	74	After refinement procedure otherwise even worse result.
	p = 4	3.60	(1,6,7,12,13,17,25,26, 31,34,35,36,39) (2,4,10,18,28, 32,37,38,40,42) (3,11,20,24,27,30) (5,8,9,14,15,16,19,21, 22,23,29,33,41,43)	(3,6,7,10, 14) (1,2,9,6) (11,12,13) (8,4,5,15)	29	79	After refinement. procedure. Without refinement step e_o = 38 and e_v = 84.
Agarwal's modified model	p = 7	<= 50	- do -	- do -	- do -	- do -	merging step and refinement procedure.
	p = 5	<= 50	same as Kusiak's p =5	same as Kusiak's p =5	39	74	After refinement procedure.
Proposed model (MAAM)	l = 18	0.18	(1,12,13,25,26,31,39) (2,4,10,18,28,32, 37,38,40,42) (3,11,20,24,27,30) (5,8,9,14,15,16,19,21, 22,23,29,33,41,43) (6,7,17,34,35,36)	(7,10) (1,2,9,6) (11,12,13) (4,5,6,8,15) (3,4)	30	58	Best result is achievable. Cell size limit = 7. Big dimension problem is solved more. efficiently

Efficiency measures are

Efficiency measures	Kusiak's and Modified model $e_v = 79$, $e_o = 29$	Proposed model (MAAM)
grouping efficiency	84.30%	87.21%
grouping efficacy	47.31%	52.17%
global efficiency	76.98%	76.19%
group efficiency	61.33%	57.14%

(b) MWSC

Table 7.15 : Solutions of various models using MWSC for example problem 5

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 5	16.92	(1,6,7,12,13, 17,25,26,31, 34,35,36,39)	(3,6,7,10, 14)	29	79	After refinement procedure
			(2,4,10,18, 28,32,37, 38,40,42)	(1,2,9,6)			Without refinement step $e_o = 38$ and $e_v = 84$
			(3,11,20, 24,27,30)	(11,12,13)			
			(5,8,9,14,15, 16,19,21,22, 23,29,33, 41,43)	(8,4,5,15)			
Agarwal's modified model	p = 5	<= 50	no integer solution possible.				
Proposed model (MAAM)	l = 20	0.17	(1,12,13,25, 26,31,39)	(7,10)	30	58	Best result is achievable. Cell size limit = 7
			(2,4,10,18, 28,32,37, 38,40,42)	(1,2,9,6)			
			(3,11,20, 24,27,30)	(11,12,13)			Big dimension problem is solved more efficiently.
			(5,8,9,14,15, 16,19,21,22, 23,29,33, 41, 43)	(4,5,6,8,15)			
			(6,7,17,34, 35,36)	(3,4)			

Efficiency measures are the same as that for case (a).

- In case (a), Kusiak's model with p = 4 gives a better result than with p =5. In case (b), with p = 5 we get the same result of Kusiak's model with p = 4 of case (a). In the case (b) with p =5 in Kusiak's model the resulting solution forms four families.
- Proposed model forms five families and gives the best result as the value of e_o + e_v (= 88) is smaller than that for Kusiak's model with p = 4 (that is its better solution) which is 108.
- CPU time is very less as compared to that of Kusiak's model.

7.1.1.2 General observations

1. Where do we need to go through refinement procedure?

   Mostly with MWSC we need this procedure, however, if the distribution of operations is not sparse then we may not need this. It is because of the fact that high weighing factor of the machine having larger number of operations when maximized will pull that part towards itself (that is towards the cell containing this machine) which uses some other machines of low weighing factors besides the high weighing factor machines. So, even when the part has more number of operations in some other cell it will be found, as a result, in the cell having the machine of high weighing factor.

2. It is because of the refinement procedure applied to reassign some leftover bottleneck parts that final result obtained by the three models are quite near to each other.

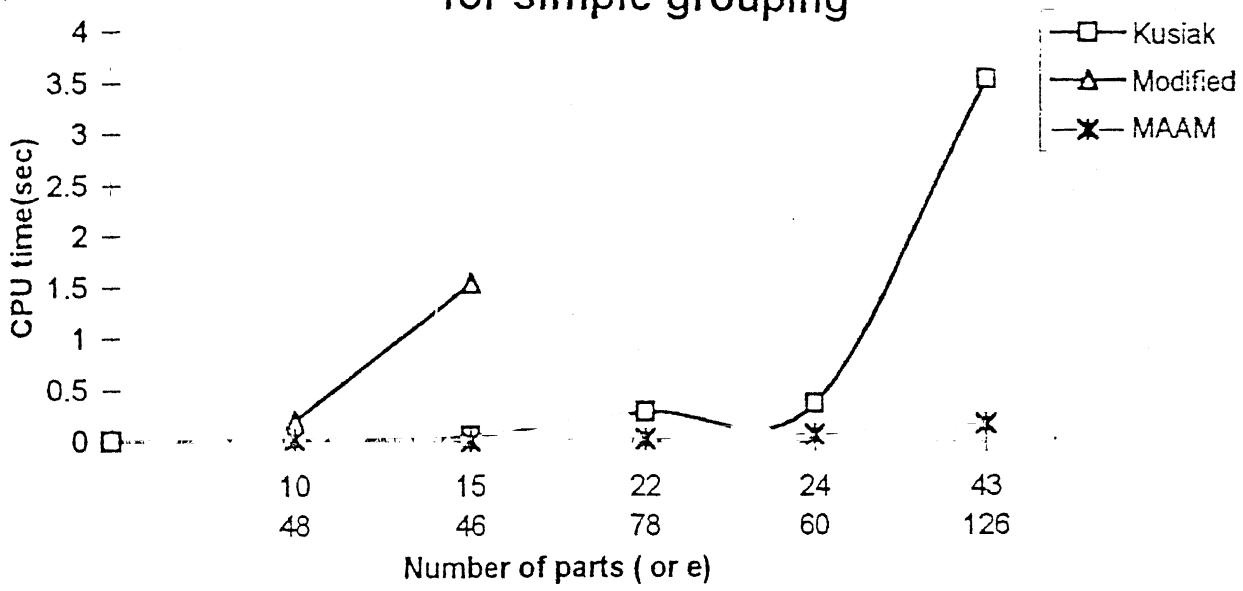
3. Use of MWSC results in less than the number of families 'p' given when p is given a higher value whereas with Lee's distance coefficient the solutions results into exactly 'p' number of families given.

4. Use of MWSC with the proposed model results in less number of loops when distribution of operations on various machines is somewhat uniform.
5. Use of MWSC in case of simple grouping can be advantageous when the distribution of operations on machines is less sparse. Otherwise, one needs to go through the refinement procedure arises.
6. Bearing the above observations in mind, use of MWSC in the case of generalized grouping is not advisable since when there is more than two alternative machine for the same operation it may result in having $\sum u_{ml} > 0$ for only two of these alternative machines, where l is the family to which machine m has been assigned. Also $\sum u_{ml}$ for one of these two will be higher than the other thus creating unnecessary intercell movement which, otherwise, by using Lee's distance coefficient would not occur.

7.1.1.3 Graphical analysis of solutions

Graph of number of plans and 'e' vs CPU time for various models is shown in Fig. 7.1.

FIGURE 7.1 :Number of parts(or e) vs CPU time for simple grouping



It is found that for the proposed model (MAAM) CPU time varies quadratically with number of plans while for Kusiak's model (PMM) it varies exponentially. No such conclusion can be drawn for Agarwal's modified model as we had to limit its time of computation for most of the example problems taken.

Graphs between problem number and CPU time for Kusiak's model and the proposed model using Lee's distance coefficient and MWSC are shown in Figs. 7.2 and 7.3.

FIGURE 7.2 : Problem number vs CPU time (Kusiak model)

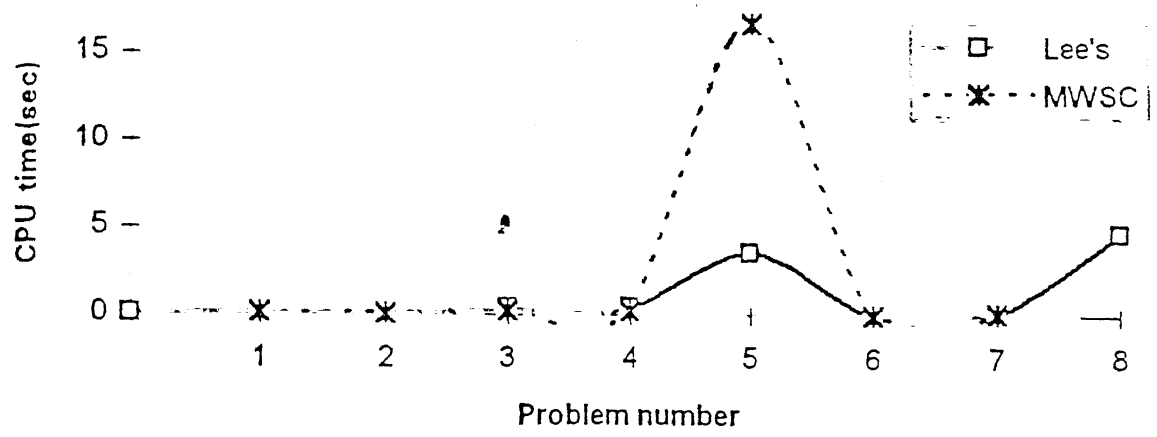
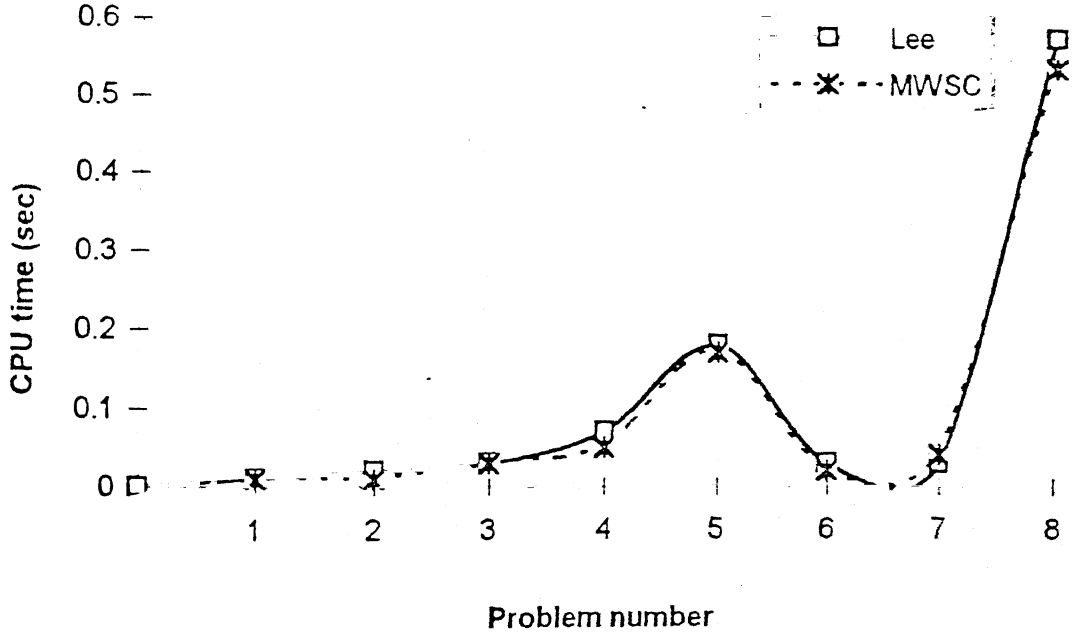


FIGURE 7.3 : Problem number vs CPU time (Proposed MAAM model)



It is observed that for Kusiak's model time variation is large when MWSC results in less number of process families than 'p', the maximum number of families required. However, for the proposed model this variation is negligible.

7.1.2 Generalized Grouping Problems

Five example problems have been taken in the class of generalized grouping problems (2 of PWRF type and 3 of OWRF type). The 0-1 input matrix for the first three example problems (2 of PWRF type and 1 of OWRF type) are shown in the following tables (Tables 4.1, 7.16 and 7.17).

Example Problem 6 (problem type PWRF)

The input data are the same as in Table 4.1.

Example Problem 7 (problem type PWRF)

Table 7.16 : Binary Input Incidence Matrix for Example Problem 7

			Machines				
Part k	Plan no. of part k	Plan refer. no. i	1	2	3	4	5
1	1	1	1	1	1	1	0
1	2	2	1	1	1	0	0
1	3	3	0	1	1	0	0
2	1	4	1	1	1	1	0
2	2	5	1	0	1	0	0
3	1	6	0	0	1	1	1
3	2	7	0	0	0	1	1
4	1	8	0	0	1	1	1
4	2	9	0	0	0	0	1
5	1	10	1	1	1	0	0
5	2	11	0	0	1	0	0
6	1	12	1	0	1	0	0
6	2	13	0	0	1	0	0
7	1	14	0	1	1	0	0
7	2	15	0	0	0	1	0

Example Problem 8 (problem type OWRF)

Table 7.17 : Binary Input Incidence Matrix for Example Problem 8

Part k	plan no. of part k	Plan refer. no. i	Machines																			
			01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
1	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
1	2	2	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
2	1	3	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2	2	4	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
3	1	5	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
3	2	6	1	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
4	1	7	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	2	8	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
5	1	9	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
5	2	10	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0
5	3	11	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
6	1	12	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	13	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	14	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0
7	2	15	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0
7	3	16	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
7	4	17	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
7	5	18	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
7	6	19	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	20	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
8	2	21	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
9	1	22	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0
9	2	23	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0
9	3	24	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0
9	4	25	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0
9	5	26	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
9	6	27	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
10	1	28	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
10	2	29	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
11	1	30	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0
11	2	31	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
11	3	32	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
12	1	33	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
12	2	34	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
12	3	35	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
13	1	36	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0
13	2	37	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
13	3	38	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1
14	1	39	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	0	0
14	2	40	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	0
14	3	41	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1
15	1	42	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0
15	2	43	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0
15	3	44	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

16	1	45	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0
16	2	46	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0
16	3	47	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1
17	1	48	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0
18	1	49	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
19	1	50	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
20	1	51	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

7.1.2.1 Methodology, solutions and observations

As mentioned in the previous section 7.1.1.2 we do not use MWSC in the case of generalized grouping problem. Further, the proposed heuristic is applied in whole (i.e. all the steps) to the generalized grouping problems. However, in small size problem few steps may not get executed. The size of the problem is given by (plan x machine).

With the Kusiak's model we get the results as process families which is then transformed into corresponding part families. To reach at the optimal number of process families to be formed the model is run few times by trial and error (by varying the values of p). Machine assignment and refinement procedure of the proposed heuristic is also used after getting the solution from the formulation.

With the proposed model (MAAM), we get the results as process families which may not be disjoint for bigger size problem (plans x machines > 50) and thus merging step (step2) of the proposed heuristic must be applied. After that machine assignment (step 3) and refinement procedures (step 5)of the proposed heuristic are applied to assign machines to the families and to improve the solution, respectively. The final results are in terms of part families and their corresponding machines cells.

Solutions of Example Problem 6

Problem size : (11x4) and total number of operations (e') = 20

Results by the three models are the same and shown below and in the Table 7.18.

Total number of operation in the resulting solution (e) = 9.
Number of bottleneck operations (e_o) = 0 and number of voids (e_v) = 1.

Table 7.18 : Solutions of various models using Lee's distance Coefficient for example problem 6

Model	p or l	CPU time (sec)	Process families	Part families	Machine cells	Remarks
Kusiak's model (PMM)	p = 2	0.02	(2,7) (5,9,11)	(1,3) (2,4,5)	(2,4) (1,3)	
Agarwal's modified model	p = 2	0.09	- do -	- do -	- do -	
Proposed model (MAAM)	l = 2	0.03	- do -	- do -	- do -	no merging required

Note : For small size problems (plans x machines <50), Kusiak's model may take less time than the proposed model but the difference is just insignificant.

Efficiency measures are :

Efficiency measures	All models
grouping efficiency	95.00%
grouping efficacy	90.00%
global efficiency	100.00%
group efficiency	100.00%

Solutions of Example Problem 7

Problem size : (15x5), total number of operations (e′) = 34.

The results obtained by all three models are the same and are shown below and in the Table 7.19.

Total number of operation in the resulting solution (e) = 17.
Number of bottleneck operation (intercell movement) (e_0) = 1.
Number of voids over all the part families (e_v) = 3.

Table 7.19 : Solutions of various models using Lee's distance coefficient for example problem 7

Model	p or l	CPU time (sec)	Process families	Part families	Machine cells	Remarks
Kusiak's model (PMM)	p = 2	0.03	(2,5,10, 12,14) (6,8)	(1,2,5, 6,7) (3,4)	(1,2,3) (4,5)	
Agarwal's modified model	p = 2	4.07	- do -	- do -	- do -	
Proposed model (MAAM)	l = 2	0.03	- do -	- do -	- do -	merging required

Efficiency measures are :

Efficiency measures	All models
grouping efficiency	88.57%
grouping efficacy	80.00%
global efficiency	94.11%
group efficiency	88.88%

Solutions of Example Problem 8

Problem size : (51x20) with total number of operations (e′) = 180

The results obtained by all three models are shown in Table 7.20.

Total number of operations resulting in all the solutions (e) = 67

Table 7.20 : Solutions of various models using Lee's distance coefficient for example problem 8

Model	p or l	CPU time (sec)	Part families	Machine cells	e_o	e_v	Remarks
Kusiak's model (PMM)	p = 5	4.71	(1,2,3,4,5) (6,7,8,9,10) (11,12,13) (14,15,16, 17) (18,19,20)	(1,6,9,12,20) (2,5,7,16) (3,8,11,19) (10,14,17,18) (4,13,15)	2	17	no improvement by refinement procedure.
Agarwal's modified model	p = 5	<= 50	no integer solution possible	-	-	-	-
Proposed model (MAAM)	l = 7	0.58	(1,2,3,4,5) (6,7,8,9, 10) (11,12,13) (14,15, 16, 17) (18,19,20)	(1,6,9,12) (2,5,7,16, 20) (3,8,11,19) (10,14, 17, 18) (4,13,15)	1	15	best result is achievable. Cell size limit = 7 large size problems are solved more efficiently.

Efficiency measures are :

Efficiency measures	Kusiak's model and Agarwal's modified model	Proposed model
grouping efficiency	95.25%	96.00%
grouping efficacy	77.38%	80.48%
global efficiency	97.01%	98.05%
group efficiency	95.74%	97.87%

For studying the performance about the CPU time requirement, we take two more examples of OWRF type. They are given in the following tables. Observation on p or l and CPU times have been given in Tables 7.21 and 7.22.

Example Problem 9 (problem type OWRF)

Problem size : (31x10).

Number of parts	Number of plans	e′	e
10	31	131	42

Table 7.21 : Observations on p or l and CPU-time for example problem 9

Model	p or l	CPU time (sec)
Kusiak's model (PMM)	p = 3	0.57
	p = 4	0.15
Agarwal's modified model	p = 3	Disc quota reached
	p = 4	(> 50 sec)
Proposed model (MAAM)	l = 5	0.06

Example Problem 10 (problem type OWRF)

Problem size : (26x10)

Number of parts	Number of plans	e′	e
15	26	69	47

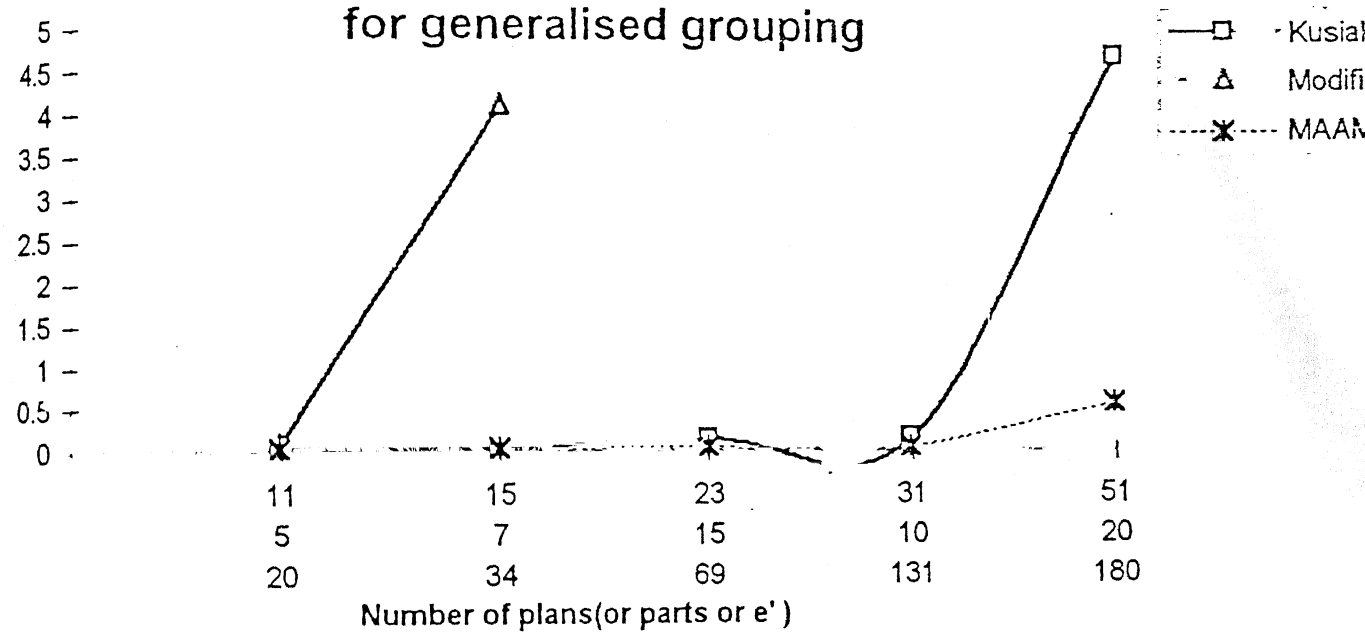
Table 7.22 : Observations on p or l and CPU-time for example problem 10

Model	p or l	CPU time (sec)
Kusiak's model (PMM)	p = 3	0.13
Agarwal's modified model	p = 3	Disc quota reached
	p = 4	(> 50 sec)
Proposed model (MAAM)	l = 5	0.05

7.1.2.2 Graphical analysis of solutions

Graph for number of plans, parts and e′ vs CPU time for all the models is shown in Fig. 7.4.

FIGURE 7.4 : Number of plans (or parts or e') vs CPU time for generalised grouping



The conclusions here are the same as in case of simple grouping problems, that is, quadratic CPU-time variation for the proposed model (MAAM), and exponential CPU-time variation for Kusiak's model.

Graphs for various example problems vs four efficiency measures for various models are shown in Figs. 7.5, 7.6, 7.7 and 7.8.

FIGURE 7.5 :Problem number vs Grouping efficiency

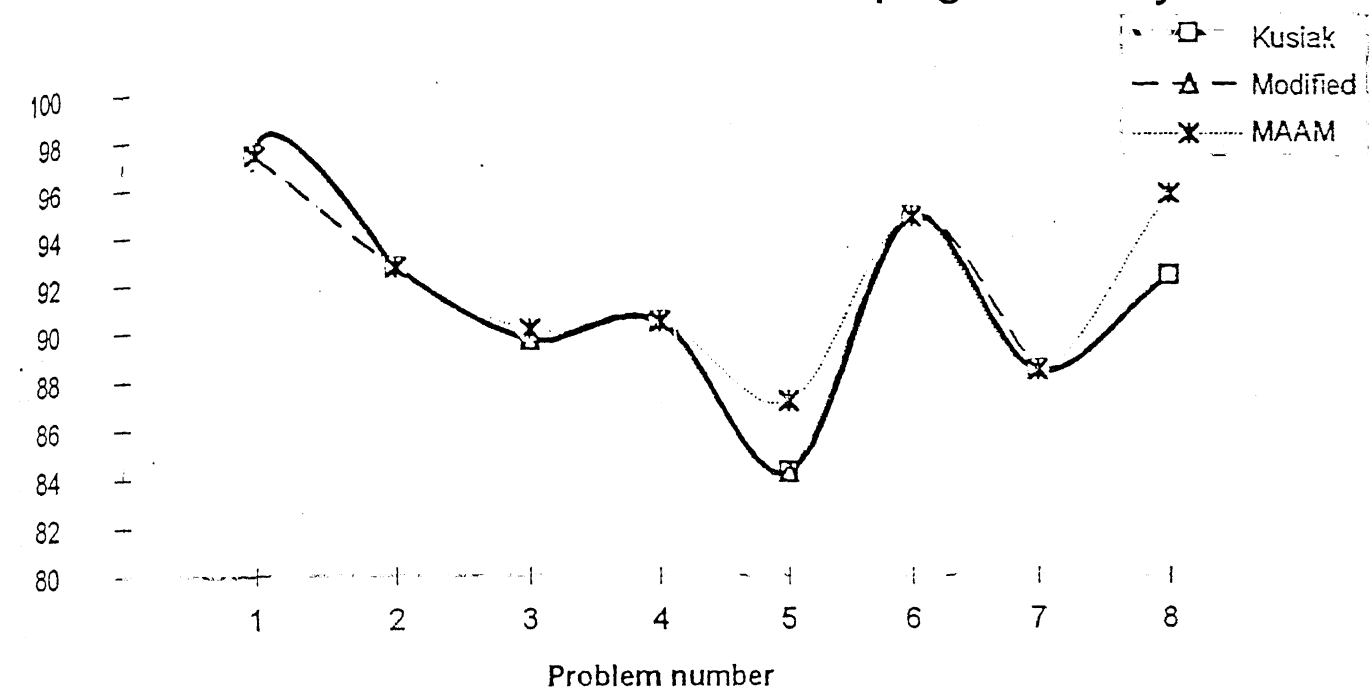


FIGURE 7.6 : Problem number vs Grouping efficacy

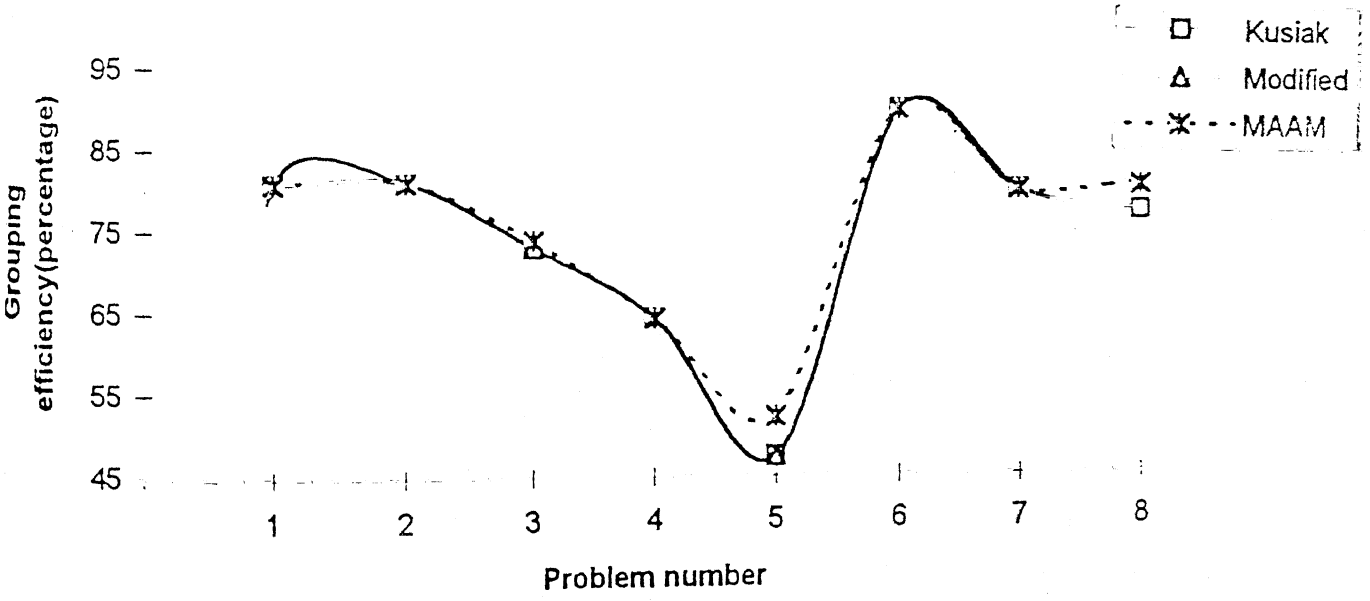


FIGURE 7.7 : Problem number vs Global efficiency

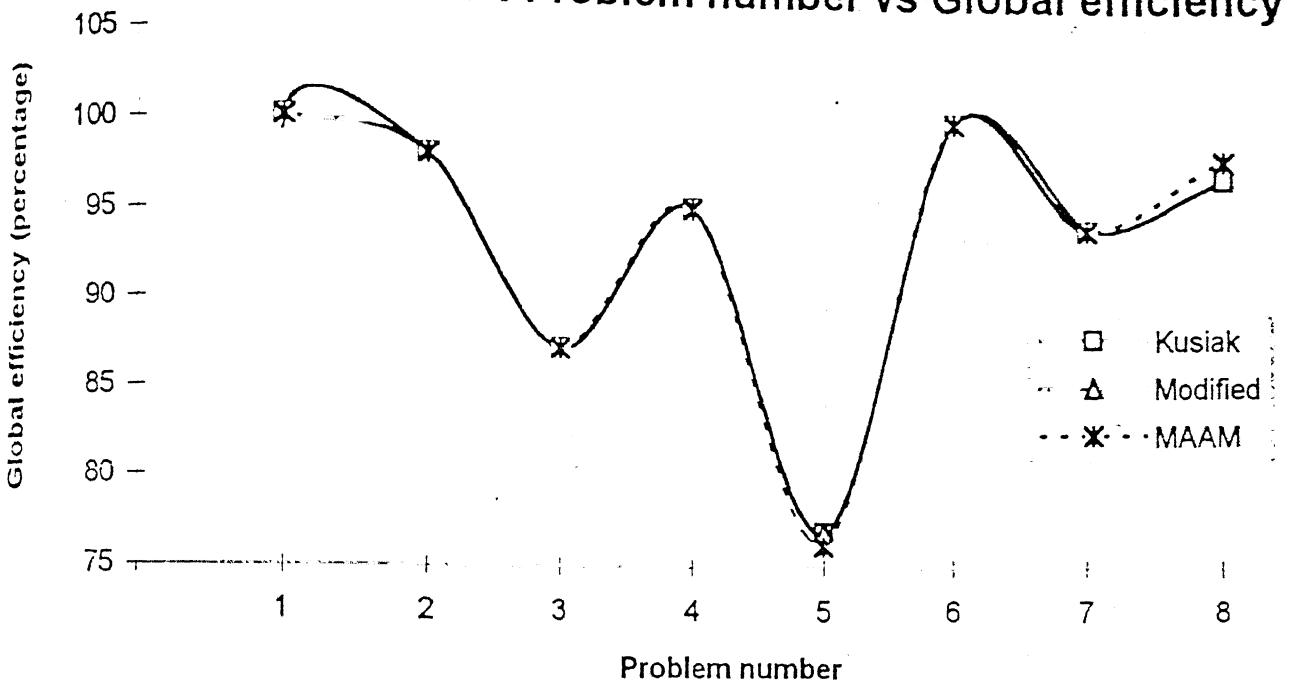
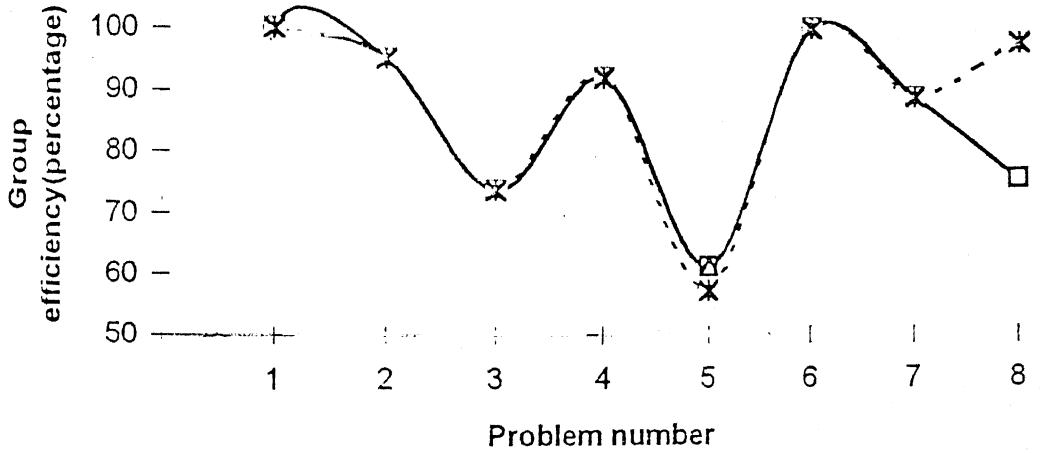


FIGURE 7.8 : Problem number vs Group technology efficiency



For most of the examples taken (simple as well as generalized grouping), all of these measures are *at least* better for the proposed model (MAAM).

7.2 VALIDATION OF PROPOSED SIMILARITY AND DISSIMILARITY COEFFICIENTS

We test some example problems on simple grouping taken from the literature having process time and operation sequence inputs and the grouping results are obtained by applying their respective coefficients WPTSC and OSMUDC, in either Kusiak's model (PMM) or the proposed model (MAAM). These results are then compared with their corresponding results from the literature.

7.2.1 Observations on Weighted Part and Time Similarity Coefficient (WPTSC)

Example Problem 11

The problem by Wu and Salvendy [1993] is shown in Table 7.23.

Table 7.23 : Process Time Part-Machine Incidence Matrix for Example Problem 11

		Machines						
		1	2	3	4	5	6	7
	1	5	1	0	0	0	9	0
P	2	3	0	0	7	0	0	0
a	3	1	3	8	0	0	0	0
r	4	0	0	0	6	0	0	5
t	5	0	0	3	0	7	0	0
s	6	0	0	10	0	11	0	0
	7	7	2	0	2	0	0	0

Wu and Salvendy applied the network flow approach for minimizing the time of intercell movement between the machine cells. They obtained first the machine cells,

and later the parts were allotted to the cells. The solutions obtained by them are given in Table 7.24.

Table 7.24 : Wu and Salvendy's solution for example problem 11

Machine cells	Part families
(4,7)	(2,4)
(1,6)	(1,7)
(2,3,5)	(3,5,6)
Total number of intercell movements	5
Total time of intercell movements	8 units

Solution by using WPTSC (with weighing factors $I_p = 5.0$, $I_t = 2.0$.) in the proposed model (MAAM) is given in Table 7.25. As expected from MAAM, we obtain first the part families, and machines are then assigned to the families to form the machine cells.

Table 7.25 : Solution by using WPTSC with MAAM for example problem 11

Part families	Machine cells
(1,3,7)	(1,6,2)
(4,2)	(4,7)
(5,6)	(3,5)
Total number of intercell movements	2
Total time of intercell movements	13 units

For this example, when we take $I_p = 2.0$, $I_t = 5.0$ we again find the same solution as given in the Table 7.25.

Experiment

Treating machines as parts and parts as machines, the result is obtained first in the form of machine cells to which parts are later allotted. Further, it is noted that applying WPTSC in this reversed case, we find exactly the same results as obtained by Wu and Salvendy (given in Table 7.24). Thus, we can say that WPTSC is an effective representation of similarity between a pair of plans, and also an effective measure to minimize the time of intercell movements.

Example Problem 12

This problem has been solved by Seth [1995] using Mosier's process time related coefficients viz., Additive Weighed Similarity Coefficient (AWSC) and Multiplicative Weighed Similarity Coefficient (MulWSC) developed by Mosier [1985].

Table 7.26 : Input Data for Example Problem 12

		Machines														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	3	0	0	0	0	0	0	0	1	3	2	0	0	0
	2	0	0	4	0	2	0	0	1	0	0	0	0	3	0	2
P	3	2	0	0	0	0	2	0	0	2	0	0	0	0	2	0
a	4	1	0	0	1	0	0	0	0	1	0	0	0	0	4	0
r	5	0	0	2	0	3	0	0	3	0	0	0	0	3	0	2
t	6	1	0	0	1	0	1	0	0	1	0	0	0	0	1	0
s	7	0	4	0	0	0	0	4	0	0	4	4	4	0	0	0
	8	0	0	3	0	3	0	0	3	0	0	0	0	3	0	3
	9	0	0	0	3	0	1	0	0	4	0	0	0	0	3	0
	10	0	3	0	0	0	0	2	0	0	4	2	1	0	0	0

Solutions by applying Mosier coefficients in Kusiak's PMM model are given in the Table 7.27.

Table 7.27 : Solution by using AWSC/MulWSC in PMM for Example Problem 12

Part families	Machine cells
(1,7,10)	(2,7,10,11,12)
(2,5,8)	(3,5,8,13,15)
(3,4,6,9)	(1,4,6,9,14)

By using WPTSC ($I_p = 5.0$, $I_t = 2.0$) in the Kusiak's model we obtained the same result as obtained by Seth (Table 7.27). Here also it is found that there is no effect of reversing the values of weighing factors I_p and I_t of part movement and time movement Thus, it can be concluded that MuWSC and TMDC are only different representation of

the same kind of similarity between the parts. However, it can be noted that this conclusion is problem specific.

7.2.2 Observations on OSMUDC

Example problem 13

The problem is shown in Table 7.28 with total number of operations (e) = 78.

Table 7.28 : Input Data for Example Problem 13

Number of parts	20
Number of machines	20
Part number	Sequence of machines required
1	12 1 9 18 20
2	11 3 2
3	8 20 19
4	3 11 2 10
5	4 15 6 7
6	11 14 16 17 5
7	5 16 17
8	15 13 7 9 4
9	18 9 11 1 12
10	19 20 8
11	11 14 3
12	9 18 5 12 1
13	6 7 15 17
14	8 10 12
15	13 14 16 17
16	15 7 6 19
17	9 1 12
18	8 19 20 10
19	3 2 11 5
20	18 10 1 12

Seth [1995] obtained the solution by using SLBSC [Choobineh, 1988] in Kusiak's model for part families formation and the machine cell formation in that order. The solution is shown in Table 7.29.

Table 7.29 : Solution using SLBSC in Kusiak's model

Machine cells	Part families
(2,3,5,11,13,14,16,17)	(2,4,6,7,11,15,19)
(1,9,10,12,18)	(12,1,17,9,20,14)
(8,19,20)	(3,10,18)
(4,6,7,15)	(5,8,13,16)
Total number of intercell movements = 10.	

Efficiency measures are :

Global efficiency	87.18%
Group efficiency	81.13%
G.T.efficiency	75.86%

Now, the results are obtained by using OSMUDC (with inputs $w_b = 1.0$, $w_o = 5.0$ $w_i = 2.0$, and UP_m = number of operations under machine machine m) in Kusiak's model (with p = 4) and applying steps 3 and 5 of the heuristic proposed are shown in Table 7.30.

Table 7.30 : Solution by using OSMUDC in Kusiak's model

Part families	Machine cells
(2,4,6,7,11,15,19)	(2,3,5,11,14,16,17)
(1,9,12,14,17,20)	(1,9,10,12,18)
(3,10,18)	(8,19,20)
(5,8,13,16)	(4,6,7,15,13)
Total number of intercell movements = 10.	

Efficiency measures are :

Global efficiency	87.18%
Group efficiency	81.13%
G.T.efficiency	77.58%

An improvement in G.T. efficiency is observed because machine 13 is kept in a different cell, however, that does not affect other efficiencies as the total number of intercell movements in both the cases remains the same.

7.3 TESTING OF PROPOSED IMPROVEMENTS IN KUSIAK'S MODEL (PMM)

The testing is performed by solving some of the example problems of sec. 7.1.1.1 using Lee's distance coefficient in the improved models : IMPGRP and NATGRP. The purpose of the improvement that at least two parts (plans of different parts) should always be included in a process family, is also observed. A comparison of CPU time taken by Kusiak's model, IMPGRP model, and NATGRP model for given values of p and B (the maximum number of parts desired to be kept together in a family) is also performed. We take example problems 1, 2, 3, 4 and 5 (of simple grouping type) and 8 (generalized grouping of OWRF type) for this purpose.

7.3.1 Observation on Part Family Formation and CPU Time

Observations on part family formation and CPU time are shown in Tables 7.31, 7.32, 7.33, 7.35, 7.36 and 7.37. Table 7.34 shows the results of experimentation with varying the values of B on example problem 3.

Observations on Example Problem 1

Table 7.31 : Solutions of PMM, IMPGRP and NATGRP models for example problem 1

Model	B	p or l	CPU time (sec)	Single part (plan) family
Kusiak's model (PMM)	-	p = 4	0.04	Yes, part number 7
IMPGRP	B = 10	p = 4	0.06	No
NATGRP	B = 10	l = 6	0.08	No

Observations on Example Problem 2

Table 7.32 : Solutions of PMM, IMPGRP and NATGRP models for example problem 2

Model	B	p or l	CPU time (sec)	Single part (plan) family
Kusiak's model (PMM)	-	p = 4	0.02	Yes, part number 2
IMPGRP	B = 5 or 10	p = 4	0.01	No
NATGRP	B = 5 or 10	l = 4	0.01	No

Observations on Example Problem 3

Table 7.33 : Solutions of PMM, IMPGRP and NATGRP models for example problem 3

Models	B	p or l	CPU time (sec)	Single part (plan) family
Kusiak's model (PMM)	-	p = 6	0.11	Yes, part number 8
IMPGRP	B = 6	p = 6	0.11	No
NATGRP	B = 6	l = 8	0.06	No

Table 7.34 : CPU-time Experimentation with B in IMPGRP and NATGRP models

	CPU time (sec)	
B	IMPGRP model	NATGRP model
4	(> 50 sec)	0.06
5	(> 50 sec)	0.17
6	0.11	0.06
7	0.12	0.05
8	1.59	0.09
9	0.20	0.09
10	0.44	0.24
11	0.26	0.14
12	(> 50 sec)	0.12
13	,,	0.13
14	,,	0.12
15	,,	0.10
16	,,	0.13
17	,,	0.13
18	,,	0.06
19	,,	0.13
20	,,	0.10
21	,,	0.09
22	,,	0.15

We observe from the above table that for NATGRP model (i.e. in case of natural grouping), the variation in CPU time with respect to values of B is negligible as compared to those with IMPGRP model (i.e. in case of imposed grouping). Hence, any

reasonable value of B can be used in the case of NATGRP model while for IMPGRP model, a value of B needs to be explored for a minimum CPU time to result. This is so because of imposing the constrain of number of families in IMPGRP model.

Observations on Example Problem 4

Table 7.35 : Solutions of PMM, IMPGRP and NATGRP models for example problem 4

Models	B	p or l	CPU time (sec)	Single part (plan) family
Kusiak's model	-	p = 6	0.25	Yes, part number 6
IMPGRP model	B = 5,7 or 10	p = 6	(>30 sec)	-
NATGRP model	B = 7	l = 11	0.07	No

Observations on Example Problem 5

Table 7.36 : Solutions of PMM, IMPGRP and NATGRP models for example problem 5

Models	B	p or l	CPU time (sec)	Single part (plan) family
Kusiak's model	-	p = 7	9.41	No
IMPGRP model	B = 10	p = 7	(> 50 sec)	-
NATGRP model	B = 10	l = 18	0.98	No

Observations on Example Problem 8

Table 7.37 : Solutions of PMM, IMPGRP and NATGRP models for example problem 8

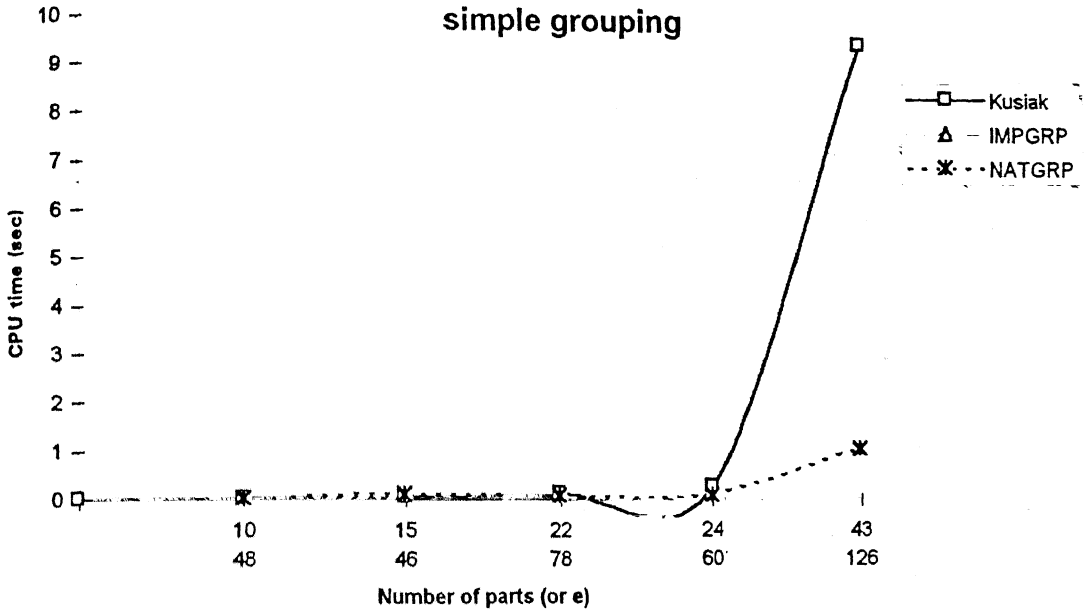
Models	B	p or l	CPU time (sec)	Single part (plan) family
Kusiak's model	-	p = 5	5.44	No
(PMM)		p = 7	18.93*	No
IMPGRP model	10	p = 5	(> 50 sec)	-
		p = 7	2.60	No
NATGRP model	5	l = 8	1.17	No
	10	l = 8	1.81	No

* Larger value of 'p' in Kusiak's model not necessarily always results in lesser CPU time as can be observed in the above table.

7.3.2 Graphical Analysis of Solutions

Graph of total number of plans and e' vs CPU time is shown in Fig. 7.9.

FIGURE 7.9 : Number of parts (or e) vs CPU time for simple grouping



It is found that with NATGRP model, CPU time varies quadratically with the total number of plans while that with Kusiak's model it varies exponentially. However, IMPGRP is not able to solve the problem of large number of plans within permissible time. These observations validate their comparisons already given in the Table 4.2 on the basis of the total number of enumerations.

Some interesting findings about the similarity between the pair of models are worth to be noted. It is seen that NATGRP and the proposed models are somewhat similar with respect to solutions CPU times and the final solutions as they both performs natural grouping. The CPU times for the case of NATGRP model are, however, little higher

than that for the case of proposed MAAM model. This is due to larger number of total constraints (2q + K) in case of NATGRP model than that in the case of the proposed model (2K + q) [see Table 4.4]. while Agarwal's modified model and IMPGRP model are similar in the way of grouping i.e. imposed grouping and very high CPU time, their differences are the violation of GT concept by Agarwal's modified model and the value of B. This value of B is the maximum number of parts which are desired to be kept together in a family for the IMPGRP model and (K-p+1) for Agarwal's modified model.

8

CONCLUSIONS AND SCOPE OF FUTURE RESEARCH

7.1 CONCLUSIONS

In the present work, a new integer programming model (MAAM) has been presented for clustering parts to form part families in the generalized grouping environment where each part can have alternative process plans and their operations can be performed on alternative machines. The proposed model (MAAM) is shown to be equivalent to unit capacitated circulation network flow (UCCNF) model with the objective of minimizing the cost of flow. The resulting solution is in the form of a main loop and several sub-loops which are interpreted as process families. The model does not require any knowledge about the number of families to be formed a priori which is one of the requirements in the case of p-median model (PMM) of Kusiak[1987]. The model ensures that at least two parts will be contained in a family while Kusiak's model does not. Singleton family i.e. family of single part may get formed by Kusiak's model. The model produces the number of families by itself. However, as the families formed are large in number and also not disjoint of each other for large problem, so, a simple heuristic procedure is proposed to reach to the optimal solution. In fact, part of the heuristic is common to the Kusiak's model also. The additional steps of the heuristic which is required for the resulting solution from the proposed model

(MAAM) is of $O(l^2)$, where l is the number of loops resulting in the solution. The proposed model itself has a low order of complexity that is of quadratic order, in terms of total number of feasible enumerations. Hence the overall complexity is also of quadratic order only, while for Kusiak's model it is of the exponential order because of the prespecified number of part-families. Agarwal's modified model has also been included for the purpose of comparing the results with the proposed model. It has been observed that the proposed model (MAAM) is computationally extremely efficient as compared to both Kusiak's model and Agarwal's modified model. The proposed model (MAAM) gives the optimal solution in one go while Kusiak's models need to be solved few times. Agarwal's modified model is not able to solve even medium size problem in the permissible time. It has been found that the heuristic procedure when applied to the solution resulting from the proposed model (MAAM) results into the best achievable solution with respect to the minimization of intercell movements. In case of Kusiak's model, application of the refinement procedure of the heuristic may result into the best achievable solution depending upon the prespecified number of families to be formed. If it is what comes as a result of applying the proposed model (MAAM) and the heuristic, then the resulting solution of the Kusiak's is the best achievable one.

Various efficiency measures are calculated for the solutions of these models. It has been found that for the proposed model (MAAM), these measures responded at least better than other models, for most of the problem taken from the literature.

Improvements in Kusiak's model have also been carried out. These are : Imposed grouping model (IMPGRP) and natural grouping model (NATGRP). They ensure that at least two part is contained in a part family which is otherwise violated by Kusiak's

model. However, we require 'B', the maximum number of parts making a part family, as an input in these improved models. This number has a major effect in IMPGRP model for the solution to result in the least computational time, but for NATGRP model it is almost ineffective. The NATGRP model is found to have similarity with the proposed model (MAAM). This has also low order of complexity (see Table 4.2) and the resulting solution need to go through the proposed heuristic. IMPGRP also need to go through some steps of the heuristic proposed.

Some measures of similarity and dissimilarity have been developed to arrest the similarity between the plans having inputs other than 0-1 nature, like, processing time of operations and their sequence. Some examples of the literature have been tested and compared with their existing solutions. Thus the validation of the developed coefficients is carried out.

7.2 SCOPE OF FUTURE RESEARCH

The proposed model (MAAM) can be tested with larger size problem including say 500 machines and thousands of parts. The problem can be created by random number generation programming. Generalized grouping with operation sequence can be viewed in such a way that the operation sequence itself can be included in the network configuration. The proposed model can be extended considering the required number of part families to be known in advance. A clue in this direction can be given as extending the dimension of variable x_{ij} of 2 to 3 i.e., x_{ijk}, where k is the part number. A new scope can be the consideration of capacity constraints of machines in either some type of measures or in the formulation itself. The clustering problem under random product demand is a new avenue for research. As the GT problem is NP-hard, so some

robust heuristic approaches like Genetic Algorithm, simulated annealing method, tabu search etc. can well be examined. Other new tools like fuzzy logic, neural nets can also be explored for their application in group technology problems.

REFERENCES

1. Agarwal, A.K., 1990, Generalized grouping and loading problems in FMS : models and methodologies, Ph.D. dissertation, Industrial and Management Engg. Dept., Indian Institute of Technology, Kanpur.
2. Ahuja, R.K., Orlin, James. B., Magnanti, Thomas. L., 1995, Network Flow, Theory, Algorithms, and Applications. (Prentice Hall, Upper Saddle River, New Jersey.).
3. Al-Qattan, I., 1990, Designing flexible manufacturing cells using a branch and bound method, International Journal of Production Research, 28, 325.
4. Askin, R.G., and Chiu, K.S., 1990, A graph partitioning procedure for machine assignment and cell formation in group technology. International Journal of Production Research, 28, 1555.
5. Askin, R.G., Cresswell, S.H., Goldberg, J.B and Vakharia, A.J., A hamiltonian path approach to recording the part machine matrix for cellular manufacturing, International Journal of Production Research, 29, 1081.
6. Ballakur, A. and Steudel H.J., 1987, A within cell utilization based heuristic for cellular manufacturing systems, International Journal of Production Research, 25, 639.
7. Boctor, F.F., 1991, A linear formulation of the machine part cell formation problem, International Journal of Production Research, 29, 343.
8. Boe, W.J., and Cheng, C.H., A close neighbour algorithm for designing cellular manufacturing systems, International Journal of Research, 29, 2097.
9. Bulbridge, J.L., 1973, A manual method of production flow analysis, Production Engineer 56, 34.
10. Carrie, A.S., 1973, Numerical taxonomy applied to group technology and plant layout, International Journal of Production Research, 11, 399.
11. Chan H.M., and Milner, D.A., 1982, Direct clustering algorithm for group formation in cellular manufacturing, Journal of manufacturing Systems, 1, 65.
12. Chandrasekharan, M.P., and Rajagopalan, R., 1986a, An ideal seed non-hierarchical clustering for cellular manufacturing. International Journal of Production Research, 24, 451.
13. Chandrasekharan, M.P., and Rajagopalan, R., 1986b, MODROC: An extension of rank order clustering for group technology, International Journal of Production Research, 24, 1221.
14. Chandrasekharan, M.P., and Rajagopalan, R., 1987, ZODIAC: An algorithm for concurrent formation of part families and machine cells, International Journal of Production Research, 25, 835.
15. Chandrasekharan, M.P., and Suresh Kumar, C., 1990 Grouping efficacy: A quantitative criterion for goodness of block diagonal forms of binary matrices in group technology, International Journal of Production Research, 28, 233.
16. Choobineh, F., 1988, A framework for the design of cellular manufacturing system, International Journal of Production Research, 26,1475.
17. Chow and Hawalaska, 1993, Intercellular part movement minimization in cell formation, International Journal of Production research, 31, 2161.
18. Chu, C.H., and Hayya, J.C., 1991, A fuzzy clustering approach to manufacturing cell formation, International Journal of Production Research, 29, 1475.
19. Chu, C.H., and Tsai, M., 1990, A comparison of three array based techniques for

manufacturing cell formation, International Journal of Production Research, 28,1417.
20. De beer, C., and DE witte, J., 1978, Production floe synthesis, Ann. CIRP, 27, 389.
21. De witte., 1980, The use of similarity coefficients in production flow analysis, International Journal of Production Research, 18 503.
22. Duda, R.O., and Hart, P.E., 1973, Pattern Classification and Scene Analysis. (John Willey & Sons.).
23. El Essawy, I.G.K., and Torrence, J., 1972, Component Flow Analysis - an effective approach to production system design, Production Engineering, 51, 165.
24. Gunasingh, K.R., and Lashkari, R.s., 1989a, Machine grouping problem in cellular manufacturing systems - an integer programming approach, International Journal of Production Research, 27, 1465.
25. Gupta, T., and Seifoddini, H., 1990, Production data based similarity coefficient for machine component grouping decisions in the design of cellular manufacturing system International Journal of Production Research, 28, 1247.
26. Harhalakis, G., Nagi, R., and Proth, J.M., 1990, An efficient heuristic in manufacturing cell formation for group technology applications, International Journal of Production Research, 28, 185.
27. Khator, S.K., and Irani, S.A., 1987, Cell formation in group technology a new approach, Computer and Industrial Engineering, 12, 131.
28. King, J.R., 1980, Machine component grouping in production flow analysis: an approach using a rank order clustering algorithm, International Journal of Production Research, 18, 213.
29. King, J.R., and Nakornchai, V., 1982, Machine component group formation in group technology review and extension, International Journal of Production Research, 20, 117.
30. Kusiak, A., 1987a, The generalized group technology concept, International Journal of Production Research, 25,561.
31. Kusiak, A., 1988, EXGT-s: A knowledge based system for group technology, International Journal of Production Research, 26, 887.
32. Kusiak, A., 1991, Branching algorithm for solving the group technology problem, Journal of Manufacturing System, 10,332.
33. Kusiak, A., and Cho, M., 1992, Similarity coefficient algorithms for solving the group technology problem, 30, 2633.
34. Lee, H., and Gracia-Diaz, A., 1993, A network flow approach to solve clustering problems in group technology, International Journal of Production Research, 31, 603.
35. Logendaran R., 1991, Impact of sequence of operations and layout of cells in cellular manufacturing. International Journal of Production Research, 29, 375.
36. Mcauley, J., 1972, Machine grouping for efficient production, Production Engineer, 51, 53.
37. Mosier, C., 1989, An experiment investigating the application of clustering and similarity coefficient, International Journal of Production Research, 27, 1811.
38. Mosier, C., and Taube, L., 1985, Weighted similarity measures heuristic for the group technology machine clustering problem, *Omega*, 13, 577.
39. Nagi, R., Harhalakis, G., and Proth, J.M., 1990, Multiple routeing and capacity considerations in group technology applications, International Journal of Production Research, 28, 2243.

40. Okabba and Chen, 1992, Intercell flow reduction heuristic, 30, 1101.
41. Onyeagoro, E.A., 1995, Group technology cell design: A case study, Production Planning and Control, 6, 365.
42. Purcheck, G.F.K., 1975b, A linear programming method for the combinatoric grouping of an incomplete power set, J. Cybernetics, 5, 51.
43. Rajagopalan, R., and Batra, J.L., 1975, Design of cellular production systems- a graph theoretic approach, International Journal of Production Research, 13, 567.
44. Riebero and Pradin, 1993, A methodology for CMS, International Journal of Production Research, 31, 235.
45. Robinson, d., and Duckstein, l., 1986, Polyhedral dynamics as a tool for machine-part group formation, International Journal of Production Research, 24, 1225.
46. Seifoddini, H., 1990, A probabilistic model for machine cell formation, Journal of manufacturing systems, 9, 69.
47. Seth, R., 1995, Comparison of cell formation techniques, M.Tech Thesis, I.M.E. Dept, I.I.T.Kanpur.
48. Shafer, S.M., and Rogers, D.F., 1993, Similarity and distance measures for cellular manufacturing. Part I. A survey, International Journal of Production Research, 31, 1133.
49. Shafer, S.M., and Rogers, D.F., 1993, Similarity and distance measures for cellular manufacturing. Part II. An extension and comparison, International Journal of Production Research, 31, 1315.
50. Shtub, A., 1989, Modeling group technology cell formation as a generalized assignment problem, International Journal of Production Research, 27, 775.
51. Song, S., and Hitomi, K., 1992, GT cell formation for minimizing the intercell parts flow, International Journal of Production Research, 30, 2737.
52. Sridhar and Suresh, 1992, A neural network approach in cell formation of group technology, International Journal of Production Research, 30, 1353.
53. Srinivasan, G., 1994, A clustering algorithm for machine cell formation in group technology using minimum spanning trees, International Journal of Production Research, 32, 2149.
54. Srinivasan, G., Narendran, T.T., and Mahadevan, B., 1990, An assignment model for the part families in group technology, International Journal of Production Research,28, 145.
55. Tam, K.Y., 1990, An operation sequence based similarity coefficient for part families formulations, Journal of manufacturing system, 5, 181.
56. Vakharia, A.J., and Wemmerlov, U., 1990, Designing a cellular manufacturing system material flow approach based on operation sequences, International Journal of Production Research, 22, 84.
57. Vanelli, A., and Kumar, R., 1986, A method for finding minimal bottleneck cells for grouping part- machine families, International Journal of Production Research, 24, 387.
58. Vohara, T., Chen, D.S., Chang, J.C., and Chen H.C., 1990, A network approach to cell formation in cellular manufacturing, International Journal of Production Research, 28, 2075.
59. Waghodekar, P.H., and Sahu, S., 1984, MACE: Machine component cell formation in group technology, International Journal of Production Research, 22, 937.
60. Wei, J.C., and Gaither, N., 1990, An optimal model for cell formation decisions, Decisions sciences, 21, 416.

61. Wei, J.C., and Kern, G.M., 1989, Commonality analysis : A linear clustering algorithm for group technology, International Journal of Production Research, 27, 2053.
62. Wemmerlov, U., and Hyer, N.L., 1987, Research issues in cellular manufacturing, International Journal of Production Research, 25, 413.
63. Wu Naiqi, and Salvendy G., 1993, A modified network approach for the design of cellular manufacturing systems.
64. Xu, H., and Wang, H.P., 1989, Part family for GT applications based on fuzzy mathematics, International Journal of Production Research, 27, 1637.

APPENDIX A

Explanation of Constarints for Example Problem

The constraints for the example problem given in Table 4.1 for the case of the proposed model (MAAM) are explained as below.

Constraints of eq. (4.4)

$x_{14} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} + x_{110} + x_{111} + x_{24} + x_{25} + x_{26} + x_{27} + x_{28} + x_{29} + x_{210} + x_{211} + x_{34} + x_{35} + x_{36} + x_{37} + x_{38} + x_{39} + x_{310} + x_{311} = 1$

$x_{41} + x_{42} + x_{43} + x_{46} + x_{47} + x_{48} + x_{49} + x_{410} + x_{411} + x_{51} + x_{52} + x_{53} + x_{56} + x_{57} + x_{58} + x_{59} + x_{510} + x_{511} = 1$

$x_{61} + x_{62} + x_{63} + x_{64} + x_{65} + x_{68} + x_{69} + x_{610} + x_{611} + x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{78} + x_{79} + x_{710} + x_{711} = 1$

$x_{81} + x_{82} + x_{83} + x_{84} + x_{85} + x_{86} + x_{87} + x_{810} + x_{811} + x_{91} + x_{92} + x_{93} + x_{94} + x_{95} + x_{96} + x_{97} + x_{910} + x_{911} = 1$

$x_{101} + x_{102} + x_{103} + x_{104} + x_{105} + x_{106} + x_{107} + x_{108} + x_{109} + x_{1101} + x_{1102} + x_{1103} + x_{1104} + x_{1105} + x_{1106} + x_{1107} + x_{1108} + x_{1109} = 1$

Constraints of eq. (4.5)

$x_{41} + x_{51} + x_{61} + x_{71} + x_{81} + x_{91} + x_{101} + x_{1101} + x_{42} + x_{52} + x_{62} + x_{72} + x_{82} + x_{92} + x_{102} + x_{1102} + x_{43} + x_{53} + x_{63} + x_{73} + x_{83} + x_{93} + x_{103} + x_{1103} = 1$

$x_{14} + x_{24} + x_{34} + x_{64} + x_{74} + x_{84} + x_{94} + x_{104} + x_{1104} + x_{14} + x_{25} + x_{35} + x_{65} + x_{75} + x_{85} + x_{95} + x_{105} + x_{1105} = 1$

$x_{16} + x_{26} + x_{36} + x_{46} + x_{56} + x_{86} + x_{96} + x_{106} + x_{1106} + x_{17} + x_{27} + x_{37} + x_{47} + x_{57} + x_{87} + x_{97} + x_{107} + x_{1107} = 1$

$x_{18} + x_{28} + x_{38} + x_{48} + x_{58} + x_{68} + x_{78} + x_{108} + x_{1108} + x_{19} + x_{29} + x_{39} + x_{49} + x_{59} + x_{69} + x_{79} + x_{109} + x_{1109} = 1$

$x_{110} + x_{210} + x_{310} + x_{410} + x_{510} + x_{610} + x_{710} + x_{810} + x_{910} + x_{111} + x_{211} + x_{311} + x_{411} + x_{511} + x_{611} + x_{711} + x_{811} + x_{911} = 1$

Constraints of eq. (4.6)

$x_{14} + x_{41} + x_{15} + x_{51} + x_{16} + x_{61} + x_{17} + x_{71} + x_{18} + x_{81} + x_{19} + x_{91} + x_{110} + x_{101} + x_{111} + x_{1101} = 0 \text{ or } 2$

$x_{24} + x_{42} + x_{25} + x_{52} + x_{26} + x_{62} + x_{27} + x_{72} + x_{28} + x_{82} + x_{29} + x_{92} + x_{210} + x_{102} + x_{211} + x_{1102} = 0 \text{ or } 2$

$x_{34} + x_{43} + x_{35} + x_{53} + x_{36} + x_{63} + x_{37} + x_{73} + x_{38} + x_{83} + x_{39} + x_{93} + x_{310} + x_{103} + x_{311} + x_{1103} = 0 \text{ or } 2$

$x_{41} + x_{14} + x_{42} + x_{24} + x_{43} + x_{34} + x_{46} + x_{64} + x_{47} + x_{74} + x_{48} + x_{84} + x_{49} + x_{94} + x_{410} + x_{104} + x_{411} + x_{1104} = 0 \text{ or } 2$

$x_{61} + x_{16} + x_{62} + x_{26} + x_{63} + x_{36} + x_{64} + x_{46} + x_{65} + x_{56} + x_{68} + x_{86} + x_{69} + x_{96} + x_{610} + x_{106} + x_{611} + x_{1106} = 0 \text{ or } 2$

$x_{71} + x_{17} + x_{72} + x_{27} + x_{73} + x_{37} + x_{74} + x_{47} + x_{75} + x_{57} + x_{78} + x_{87} + x_{79} + x_{97} + x_{710} + x_{107} + x_{711} + x_{1107} = 0 \text{ or } 2$

$x_{81} + x_{18} + x_{82} + x_{28} + x_{83} + x_{38} + x_{84} + x_{48} + x_{85} + x_{58} + x_{86} + x_{68} + x_{87} + x_{78} + x_{810} + x_{108} + x_{811} + x_{1108} = 0$ or 2

$x_{91} + x_{19} + x_{92} + x_{29} + x_{93} + x_{39} + x_{94} + x_{49} + x_{95} + x_{59} + x_{96} + x_{69} + x_{97} + x_{79} + x_{910} + x_{109} + x_{911} + x_{1109} = 0$ or 2

$x_{101} + x_{110} + x_{102} + x_{210} + x_{103} + x_{310} + x_{104} + x_{410} + x_{105} + x_{510} + x_{106} + x_{610} + x_{107} + x_{710} + x_{108} + x_{810} + x_{109} + x_{910} = 0$ or 2

$x_{1101} + x_{111} + x_{1102} + x_{211} + x_{1103} + x_{311} + x_{1104} + x_{411} + x_{1105} + x_{511} + x_{1106} + x_{611} + x_{1107} + x_{711} + x_{1108} + x_{811} + x_{1109} + x_{911} = 0$ or 2

Thus, total number of constraints = $2K + q = 2 \times 5 + 11 = 21$.

122064

Date Slip

This book is to be returned on the date last stamped. 122064

IME-1996-M-BAR-MOD